

# Desarrollo de un visor de libro electrónico en formato “digital flipbook”



Grado en Ingeniería Multimedia

## Trabajo Fin de Grado

Autora:  
Beatriz Sánchez Luna

Tutor/es:  
Sergio Luján Mora

Julio 2021



Universitat d'Alacant  
Universidad de Alicante

# Resumen

En este Trabajo Fin de Grado, se ha desarrollado una aplicación web llamada MoonBook. Es una aplicación que permite a los usuarios crear sus propios libros, compartirlos con el resto de usuarios de la plataforma y visualizarlos en un navegador web en formato flipbook. Este formato permite que los lectores tengan la sensación de leer en papel, pero con las ventajas de una aplicación web.

Para llevarla a cabo, se han utilizado diversas tecnologías entre ellas: Angular, NodeJs y MongoDB, además de diferentes librerías de JavaScript. Antes de comenzar con el desarrollo, se llevó a cabo un análisis de las diferentes plataformas que actualmente están en uso. Así como de las librerías existentes para transformar archivos y facilitar la creación del libro animado.

Tras el estudio de la cuestión, era el momento de definir los objetivos del proyecto, realizar el análisis de requisitos y el diseño de la aplicación, dónde se optó por usar una base de datos no relacional, y una interfaz sencilla.

El siguiente paso a realizar fue la implementación, siguiendo la metodología ágil Scrum, en la que por cada iteración se incrementaban las funcionalidades de la aplicación.

Finalmente, se realizaron pruebas de validación para comprobar el correcto funcionamiento de la aplicación y se obtuvo la opinión de diferentes usuarios al utilizar la aplicación.

# Motivación, justificación y objetivo general

Desde hace algunos años, me ha interesado bastante la idea de poder realizar aplicaciones que puedan ayudar a otras personas a llevar a cabo cualquier actividad o mejorar su día a día.

Al leer las ideas disponibles para realizar el Trabajo Fin de Grado, encontré la propuesta de Sergio Luján de crear un visor de libros electrónicos conocido como “digital flipbook”. El proyecto nace a partir de la solicitud de una editorial que tiene la necesidad de disponer de una plataforma para subir sus propias publicaciones de forma digital.

Este ha sido un año difícil, en el que no hemos podido estar con nuestros seres queridos, la distancia se ha adueñado de nuestro día a día, y los buenos recuerdos nos han dado la fuerza necesaria para mantenernos con una buena autoestima. Recuerdo esas charlas entre amigos en las que hablábamos sobre los últimos libros que habíamos leído. O esas tardes de invierno, con primos y tíos, entorno a la mesa de camilla, viendo álbumes entre risas, por el paso del tiempo y bonitos recuerdos de años pasados.

De ahí nace mi motivación, al ver esta propuesta pensé que realmente, se podía crear una aplicación que por un lado pudiese servir de base para la editorial y por otro, que las personas pudieran subir sus archivos, convertirlos en un formato con apariencia de libro y que pudieran compartirlos con sus amigos y/o familiares, rompiendo así con la distancia física con la que nos encontramos en estos momentos.

Desarrollar una herramienta que pueda ayudar a dos sectores diferentes es muy interesante, ya que en realidad los dos tienen la misma necesidad, poseer un lugar dónde poder compartir contenido con otras personas de forma digital. En este proyecto podré demostrar habilidades adquiridas en Ingeniería Multimedia, puesto que tendré que tocar tanto frontend como backend y aspectos aprendidos durante estos cuatro años.

## Agradecimientos

*A mi familia y amigos por el apoyo durante todos estos años.*

*A todas las personas que se han ofrecido a probar la aplicación y me  
han ayudado aportando su opinión.*

*A mi tutor Sergio por guiarme en la realización de este trabajo.*

## Citas

*Puedo porque pienso que puedo.*

***Carolina Marín***

*La creatividad es la inteligencia divirtiéndose.*

***Albert Einstein***

## Notas

En la memoria se pueden encontrar términos en inglés propios del ámbito de la informática y de la programación: flipbook, frontend, backend, ebook, plugin, framework, mockup, sprint, viewport, models, routers, controller, login, header, body, scroll, responsive.

Estos términos se han mantenido en inglés porque no existe una traducción clara al español o, si existe, no está ampliamente aceptada. Además, con el fin de evitar una sobrecarga en la presentación del texto, se ha optado por no escribirlos en cursiva.

Las figuras y tablas que aparecen son de creación propia de la autora, a excepción de aquellas en las que se indique explícitamente una fuente.

# Índice de contenidos

|   |    |
|---|----|
| Resumen.....  | 1  |
| Motivación, justificación y objetivo general .....          | 2  |
| Agradecimientos .....                                       | 3  |
| Citas .....   | 4  |
| Notas .....   | 5  |
| Índice de contenidos .....                                  | 6  |
| Índice de figuras .....                                     | 9  |
| Índice de tablas .....                                      | 11 |
| Acrónimos .....   | 13 |
| 1. Introducción .....                                       | 14 |
| 2. Estudio de viabilidad .....                              | 15 |
| 2.1. Análisis DAFO .....                                    | 15 |
| 2.2. Análisis de riesgos .....                              | 16 |
| 3. Planificación .....                                      | 18 |
| 4. Estado de la cuestión .....                              | 19 |
| 4.1. Formato digital flipbook.....                          | 19 |
| 4.2. Estudio de plataformas existentes .....                | 20 |
| 4.2.1. Flipsnack .....                                      | 20 |
| 4.2.2. Paperturn .....                                      | 21 |
| 4.2.3. Fliphtml5 .....                                      | 23 |
| 4.2.4. Yumpu .....  | 24 |
| 4.2.5. Flip Builder – Flip PDF Professional .....           | 25 |
| 4.2.6. Conclusiones.....                                    | 26 |
| 4.3. Librerías existentes para visualizar un flipbook ..... | 26 |
| 4.3.1. Turn.js.....   | 26 |
| 4.3.2. OneBook3D .....                                      | 27 |

|        |  |    |
|--------|--|----|
| 4.3.3. | BookBlock .....  | 27 |
| 4.3.4. | Comparación .....  | 27 |
| 4.3.5. | Conclusión .....   | 28 |
| 4.4.   | Librerías existentes para transformación de formatos ..... | 28 |
| 4.4.1. | De PDF a imagen .....                                      | 28 |
| 4.4.2. | Extraer contenido archivo comprimido .....                 | 29 |
| 5.     | Objetivos .....  | 30 |
| 6.     | Metodología .....  | 31 |
| 7.     | Análisis y especificación .....                            | 35 |
| 7.1.   | Requisitos funcionales.....                                | 35 |
| 7.2.   | Requisitos no funcionales .....                            | 40 |
| 8.     | Diseño.....  | 41 |
| 8.1.   | Diseño de la persistencia.....                             | 41 |
| 8.2.   | Diseño arquitectura conceptual.....                        | 45 |
| 8.3.   | Diseño API Rest .....                                      | 45 |
| 8.4.   | Diseño arquitectura tecnológica Front/Back-end .....       | 54 |
| 8.5.   | Diseño interacción o experiencia de usuario .....          | 55 |
| 8.6.   | Diseño de interfaces.....                                  | 56 |
| 8.7.   | Diseño de pruebas y validación.....                        | 69 |
| 9.     | Implementación .....                                       | 71 |
| 9.1.   | Sprint 1 – Comienzo y librería Turn.js .....               | 71 |
| 9.2.   | Sprint 2 - Archivos PDF .....                              | 71 |
| 9.3.   | Sprint 3 – Archivos ZIP.....                               | 72 |
| 9.4.   | Sprint 4 – Backend: comienzo .....                         | 73 |
| 9.5.   | Sprint 5 – Backend: usuario y libro.....                   | 74 |
| 9.6.   | Sprint 6 – Backend: login.....                             | 75 |
| 9.7.   | Sprint 7 – Backend: subida y obtención de archivos.....    | 75 |
| 9.8.   | Sprint 8 – Frontend: comienzo .....                        | 77 |



|       |  |     |
|-------|--|-----|
| 9.9.  | Sprint 9 – Frontend: crear y visualizar libro .....  | 78  |
| 9.10. | Sprint 10 – Frontend: Perfil usuario, libros y notificaciones –Backend: notificaciones<br>79 |     |
| 9.11. | Sprint 11 – Backend y Frontend: amigos e invitaciones.....                                   | 80  |
| 9.12. | Sprint 12 – Backend y Frontend: ver libros amigos y públicos, detalles libro.....            | 82  |
| 9.13. | Sprint 13 –Backend y Frontend: últimos libros, opción ver todos y filtrar .....              | 83  |
| 9.14. | Sprint 14 –Frontend: zoom y pantalla completa en el visualizador .....                       | 83  |
| 9.15. | Sprint 15 – Frontend: responsive y accesibilidad teclado.....                                | 84  |
| 9.16. | Sprint 16 – Guía página principal y pruebas de funcionamiento .....                          | 85  |
| 10.   | Pruebas y validación.....  | 86  |
| 10.1. | Resultados cuestionario .....  | 86  |
| 10.2. | Conclusiones de la prueba de usuarios .....  | 92  |
| 11.   | Resultados .....   | 96  |
| 12.   | Conclusiones y trabajo futuro .....  | 100 |
|       | Referencias.....   | 102 |

# Índice de figuras

|   |    |
|---|----|
| Figura 1 - Análisis DAFO .....  | 15 |
| Figura 2 - Varias interfaces de Flipsnack .....                                       | 21 |
| Figura 3 - Varias interfaces de Paperturn.....  | 22 |
| Figura 4 - interfaz FLIPHTML5 .....   | 23 |
| Figura 5 - Varias interfaces YUMPU .....  | 24 |
| Figura 6 - Varias interfaces de FlipPDF Professional .....                            | 25 |
| Figura 7 - Entradas Blog TFG de Beatriz Sánchez Luna .....                            | 32 |
| Figura 8 -Tablero TFG en Trello a mitad del desarrollo.....                           | 33 |
| Figura 9 - Gráfico de tiempos en Toggle a mitad de desarrollo .....                   | 33 |
| Figura 10 - Base de datos colección Usuario.....                                      | 42 |
| Figura 11 - Base de datos colección libro.....  | 43 |
| Figura 12 - Base de datos colección Notificación.....                                 | 44 |
| Figura 13 - Base de datos colección Invitación .....                                  | 44 |
| Figura 14 - Esquema arquitectura conceptual realizado con herramienta Lucidchart..... | 45 |
| Figura 15 - Arquitectura tecnológica.....   | 55 |
| Figura 16 - Mockup 1 Página Inicio .....  | 57 |
| Figura 17 - Mockup 2 Página Inicio .....  | 57 |
| Figura 18 - Mockup 3 Página Inicio .....  | 58 |
| Figura 19 - Mockup final Página Inicio .....  | 58 |
| Figura 20 - Mockup Registro .....   | 59 |
| Figura 21 - Mockup Iniciar sesión.....  | 60 |
| Figura 22 - Mockup Página principal usuario .....                                     | 61 |
| Figura 23 - Mockup Perfil usuario: Ver mi perfil .....                                | 62 |
| Figura 24 - Mockup Perfil usuario: gestionar libros .....                             | 63 |
| Figura 25 - Mockup Perfil usuario: notificaciones.....                                | 64 |
| Figura 26 - Mockup Perfil usuario: invitaciones.....                                  | 64 |
| Figura 27 - Mockup Perfil usuario: amigos.....  | 65 |
| Figura 28 - Mockup añadir amigo .....   | 66 |
| Figura 29 - Mockups añadir libro .....  | 67 |
| Figura 30 - Mockup visualizar libro .....   | 68 |
| Figura 31 - Mockup información libro.....   | 69 |

|   |    |
|---|----|
| Figura 32 - Postman: estructura de pruebas.....   | 70 |
| Figura 33 - Captura MongoDB Compass - Usuarios .....  | 74 |
| Figura 34 - Captura código encriptar contraseña usuario.....  | 74 |
| Figura 35 - Captura código diferencia función .....   | 78 |
| Figura 36 - Diagrama añadir amigo mediante invitación .....   | 81 |
| Figura 37 - Búsqueda en MongoDB.....  | 83 |
| Figura 38 - Pregunta: ¿Conoces el formato "Flipbook" que se menciona en la descripción de esta encuesta? .....  | 87 |
| Figura 39 - Pregunta: ¿Has utilizado una aplicación parecida a la descrita anteriormente?.....  | 87 |
| Figura 40 -Pregunta: En caso de haber utilizado una, ¿De cuál se trata? .....   | 88 |
| Figura 41 -Pregunta: ¿Cómo te ha parecido la creación de un libro?.....   | 88 |
| Figura 42 - Pregunta: ¿Cómo te ha parecido el uso del visor de libros? .....  | 88 |
| Figura 43 - Pregunta: ¿Has conseguido añadir un nuevo amigo? .....  | 89 |
| Figura 44 - Pregunta: ¿Has conseguido editar los datos de un libro?.....  | 89 |
| Figura 45 - Pregunta: Con respecto al uso del color y tamaño de fuentes y botones, ¿Has tenido problemas para visualizar y utilizar de manera correcta la aplicación? ..... | 89 |
| Figura 46 - Pregunta: De forma general, ¿Qué te ha parecido el uso de la aplicación? .....  | 90 |
| Figura 47 - Pregunta: ¿Qué es lo que más te ha gustado de la aplicación? .....  | 90 |
| Figura 48 - Pregunta: ¿Qué es lo que menos te ha gustado de la aplicación? .....  | 91 |
| Figura 49 - Pregunta: Una vez probada la aplicación, ¿Utilizarías una plataforma para leer que tuviera un visor "flipbook" que simula el uso de un libro en físico?.....    | 91 |
| Figura 50 - Pregunta: En este último apartado, puedes dejar cualquier aspecto para ayudar a mejorar la aplicación .....   | 92 |
| Figura 51 - Pregunta: Edad .....  | 92 |
| Figura 52 - Resultado final: Página principal .....   | 96 |
| Figura 53 - Resultado final: Iniciar sesión y registro .....  | 96 |
| Figura 54 - Resultado final: Página principal y búsqueda .....  | 97 |
| Figura 55 - Resultado final: Crear libro y adjuntar ficheros .....  | 97 |
| Figura 56 - Resultado final: Añadir amigos y detalles libro .....   | 98 |
| Figura 57 - Resultado final: Visor de libros.....   | 98 |
| Figura 58 - Resultado final: Perfil libros, perfil amigos y perfil invitaciones .....   | 99 |
| Figura 59 - Resultado final: Perfil usuario y Perfil notificaciones.....  | 99 |

# Índice de tablas

|   |    |
|---|----|
| Tabla 1 - Planificación temporal TFG.....               | 18 |
| Tabla 2 - Comparativas librerías flipbook.....          | 27 |
| Tabla 3 - RF-01 Hacer registro .....                    | 35 |
| Tabla 4 - RF-02 Iniciar sesión.....                     | 36 |
| Tabla 5 - RF-03 Cerrar sesión.....                      | 36 |
| Tabla 6 - RF-04 Crear libro.....                        | 36 |
| Tabla 7 - RF-05 Gestionar libro.....                    | 36 |
| Tabla 8 - RF-06 Visualizar libro .....                  | 37 |
| Tabla 9 - RF-07 Ver detalles libro .....                | 37 |
| Tabla 10 - RF-08 Enviar invitación de amistad .....     | 37 |
| Tabla 11 - RF-09 Aceptar/rechazar incitación .....      | 37 |
| Tabla 12 - RF-10 Ver listado amigos .....               | 38 |
| Tabla 13 - RF-11 Eliminar amigo.....                    | 38 |
| Tabla 14 - RF-12 Ver listado invitaciones .....         | 38 |
| Tabla 15 - RF-13 Ver y editar perfil usuario.....       | 38 |
| Tabla 16 - RF-14 Ver listado notificaciones .....       | 39 |
| Tabla 17 - RF-15 Bloquear/Desbloquear usuarios.....     | 39 |
| Tabla 18 - RF-16 Ver página principal usuario.....      | 39 |
| Tabla 19 - RF-17 Ver página inicio .....                | 39 |
| Tabla 20 - RNF-01 .....                                 | 40 |
| Tabla 21 - RNF-02 .....                                 | 40 |
| Tabla 22 - RNF-03 .....                                 | 40 |
| Tabla 23 - API REST GET api/usuarios.....               | 46 |
| Tabla 24 - API REST GET api/usuarios/{id}/libros .....  | 46 |
| Tabla 25 - API REST GET api/usuarios/amigos .....       | 46 |
| Tabla 26 - API REST POST api/usuarios .....             | 47 |
| Tabla 27 - API REST POST api/usuarios/amigo .....       | 47 |
| Tabla 28 - API REST PUT api/usuarios/{id} .....         | 48 |
| Tabla 29 - API REST PUT api/usuarios/amigos/lista ..... | 48 |
| Tabla 30 - API REST DELETE api/usuarios/{id}.....       | 48 |
| Tabla 31 - API REST GET api/libros .....                | 49 |
| Tabla 32 - API REST GET api/libros/publicos .....       | 49 |

|  |    |
|--|----|
| Tabla 33 - API REST POST api/libros .....                      | 49 |
| Tabla 34 - API REST PUT api/libros/{id} .....                  | 50 |
| Tabla 35 - API REST DELETE api/libros/{id}.....                | 50 |
| Tabla 36 - API REST GET api/notificaciones.....                | 50 |
| Tabla 37 - API REST POST api/notificaciones.....               | 51 |
| Tabla 38 - API REST PUT api/notificaciones/{id} .....          | 51 |
| Tabla 39 - API REST DELETE api/notificaciones/{id} .....       | 51 |
| Tabla 40 - API REST GET api/invitaciones.....                  | 51 |
| Tabla 41 - API REST POST api/invitaciones.....                 | 52 |
| Tabla 42 - API REST PUT api/invitaciones/{id}.....             | 52 |
| Tabla 43 - API REST DELETE api/invitaciones/{id} .....         | 53 |
| Tabla 44 - API REST POST api/login .....                       | 53 |
| Tabla 45 - API REST GET api/subir/{tipo}/{nombreArchivo} ..... | 53 |
| Tabla 46 - API REST POST api/subir/{tipo}/{id} .....           | 53 |

# Acrónimos

- ABP: Aprendizaje basado en proyectos.
- ACID: Atomicity, Consistency, Isolation, Durability (Atomicidad, Consistencia, Aislamiento y Durabilidad).
- API: Application Programming Interfaces (Interfaz de programación de aplicaciones).
- CRUD: Create, Read, Update, Delete (Crear, Leer, Actualizar y Borrar).
- CSS: Cascading Style Sheets (Hojas de estilo en cascada).
- DAFO: Debilidades, Amenazas, Fortalezas, Oportunidades.
- EPUB: Electronic publication (Publicación electrónica).
- HTML: HyperText Markup Language (Lenguaje de marcas de hipertexto).
- HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertextos).
- IEEE: The Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos).
- JPEG/JPG: Joint Photographic Experts Group (Nombre de un comité de expertos).
- JS: JavaScript.
- JSON: JavaScript Object Notation (Notación de objeto de JavaScript).
- MEAN: MongoDB, Express, AngularJS y NodeJS.
- PDF: Portable Document Format (Formato de documento portátil).
- PNG: Portable Network Graphics (Gráficos para la red portátiles).
- REST: REpresentational State Transfer (Transferencia de estado representacional).
- SPA: Single Page Application (Aplicación de página única).
- SQL: Structured Query Language (Lenguaje de consulta estructurada).
- TFG: Trabajo Fin de Grado.
- TS: TypeScript.
- URI: Uniform Resource Identifier (Identificador uniforme de recursos).

# 1. Introducción

Nuestra población se encuentra cada vez más digitalizada, las tendencias van cambiando y nuestras costumbres también. Es por eso, que el sector editorial se está viendo obligado a encontrar nuevas formas de vender sus libros, y muchas de ellas están apostando por el formato digital.

En este trabajo se explica el desarrollo de MoonBook, una aplicación web que pueda servir como base a dos tipos de plataformas, por un lado, al sector editorial, que desea tener una aplicación que le permita entrar al mundo digital y al mismo tiempo, dar una solución a ese sector de usuarios que le gusta compartir contenido propio con sus amistades, como pueden ser apuntes académicos, pequeños relatos o reportajes fotográficos.

Para ello, se ha desarrollado un conversor de documentos que permita al usuario subir sus archivos para a continuación visualizarlos en un formato “digital flipbook”, formato que permite mediante un navegador web, pasar las páginas de un libro de manera interactiva simulando el uso de un libro físico.

Antes de comenzar con el desarrollo, se ha realizado un estudio del estado de la cuestión para conocer aplicaciones similares y comprobar que funcionalidades ofrecen para poder comparar entre ellas diferentes aspectos y a partir de esas conclusiones, mediante metodologías ágiles se ha definido el diseño y se ha realizado la implementación de esta aplicación web.

## 2. Estudio de viabilidad

Antes de comenzar un proyecto, da igual el ámbito en el que nos encontremos, es importante realizar un estudio de viabilidad para planificarlo correctamente, estudiar sus debilidades y fortalezas y ser conscientes del tiempo y los recursos que tenemos disponibles para llevarlos a cabo minimizando así el riesgo.

### 2.1. Análisis DAFO

Según un artículo de Recursos en Project management (1), DAFO (Debilidades, Amenazas, Fortalezas, Oportunidades) es una herramienta de análisis que nos permite conocer el estado actual en el que nos encontramos a la hora de realizar un proyecto. De forma interna nos permite identificar las debilidades y las fortalezas, y de forma externa, identificar las oportunidades y las amenazas. Podemos ver el esquema del análisis realizado en este proyecto en la Figura 1.

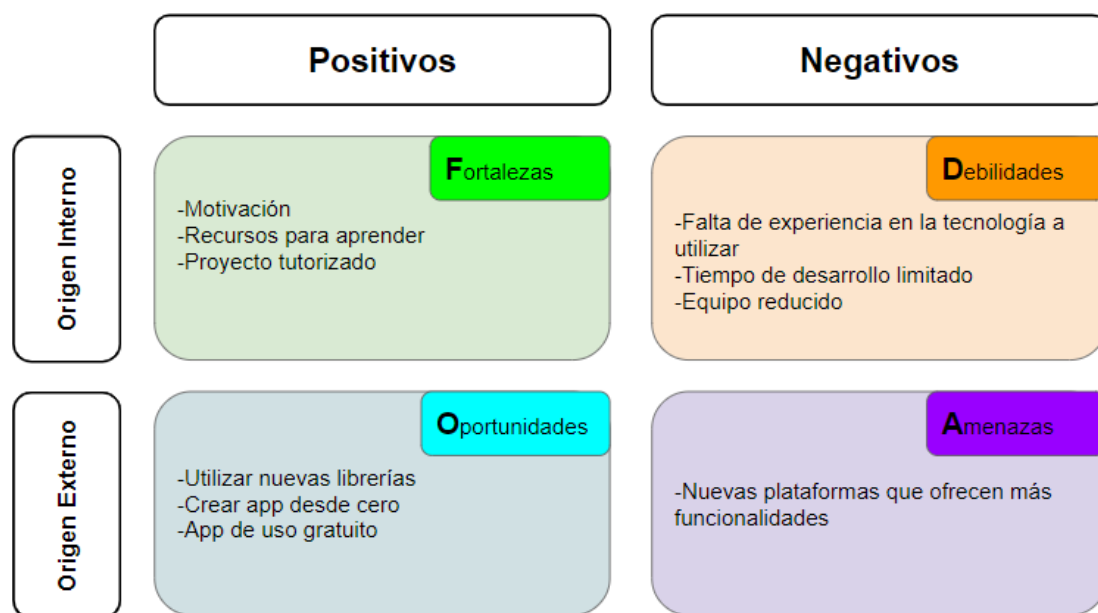


Figura 1 - Análisis DAFO

- **Fortalezas**

En fortalezas encontramos aquellas características que nos pueden facilitar conseguir aquello que estamos proponiendo, entre estas características encontramos, la motivación a la hora de realizar el proyecto, la ayuda del tutor a lo largo del desarrollo, y el uso de



tecnología como Angular y Express de las que se puede encontrar gran cantidad de documentación. Además, el desarrollo no supone un coste económico.

- **Debilidades**

En debilidades se agrupan las características que nos dificultarán a llevar a cabo el desarrollo del proyecto. Entre ellas encontramos la falta de experiencia en la transformación de formatos y poco conocimiento de librerías sobre libros electrónicos. Será necesario realizar un estudio y seleccionar las librerías más convenientes. Para terminar la aplicación al 100% de sus funcionalidades, sería necesario un mayor número de horas que no disponemos.

- **Oportunidades**

En oportunidades encontramos aquellas características que podrían facilitar el éxito del proyecto. Entre ellas encontramos la realización de una aplicación desde cero, utilizando librerías existentes. La creación de un visor de libros que no dependa de poseer dispositivos físicos concretos para leer como libros electrónicos, nos permitirá llegar a un mayor número de usuarios.

- **Amenazas**

Son aspectos externos que pueden dificultar el éxito del proyecto, como puede ser la aparición de nuevas plataformas más novedosas que ofrezcan nuevas funcionalidades, no gestionar bien el tiempo debido a la realización del proyecto ABP (Aprendizaje basado en proyectos) simultáneamente. Existencia de plataformas parecidas.

## 2.2. Análisis de riesgos

En un proyecto tan importante, es fundamental estudiar qué riesgos podemos afrontar durante el desarrollo y establecer medidas para minimizarlas. El riesgo más importante es no conseguir todos los objetivos que se plantean, por una mala gestión temporal o por no establecer unos objetivos realistas y no ser capaz de realizarlas antes de la fecha. Por ello, el desarrollo de este

proyecto se llevará a cabo poco a poco a lo largo del curso teniendo en cuenta el tiempo ocupado por el transcurso del curso. Además, dejando holgura de tiempo por posibles enfermedades o problemas con el hardware. Para no tener problemas con el software, utilizaré repositorios online para minimizar riesgos, al mismo tiempo de realizar copias en un dispositivo físico.

### 3. Planificación

Para planificar el desarrollo del proyecto, he tenido en cuenta el tiempo que voy a poder dedicarle cada mes, como el Trabajo final de grado (TFG) lo realizaré simultáneamente al resto de asignatura de cuarto curso dedicaré una media de 35/40 horas al mes, lo que implica entregar el proyecto en la convocatoria de julio para realizar el total de horas que se estiman para su desarrollo (300 horas).

Durante los primeros meses se detallará la idea que se va a llevar a cabo, se estudiarán las tecnologías a utilizar y las librerías existentes para su desarrollo. Una vez estudiadas, se especificarán las características de la aplicación y se realizará el diseño y desarrollo de manera incremental, aportando cada semana nuevas funcionalidades. Por último, se realizarán pruebas de validación y se terminará de escribir la memoria. En la Tabla 1 podemos ver un resumen de la planificación.

*Tabla 1 - Planificación temporal TFG*

| Contenidos  | Tiempo total | Fecha límite fin |
|---|--------------|------------------|
| Motivación, justificación, objetivo general, Introducción<br>Estado de la cuestión  | 1 mes        | 30 noviembre     |
| Objetivos<br>Metodología<br>Análisis y especificación<br>Presupuesto, estimaciones, planificación   | 1 mes        | 31 diciembre     |
| Diseño e implementación   | 4 meses      | 30 abril         |
| Pruebas y validación<br>Resultados<br>Conclusiones y trabajo futuro<br>Referencias, bibliografía y apéndices<br>Agradecimientos, citas, índices | 2 meses      | 30 junio         |

## 4. Estado de la cuestión

En este apartado se realizará una investigación para conocer las librerías existentes para la realización de flipbooks y plataformas que utilicen estas tecnologías para que los usuarios adjunten sus archivos y generen el libro. Una vez conocido el estado actual del mercado se podrá tomar decisiones más adecuadas para la realización del proyecto.

### 4.1. Formato digital flipbook

El objetivo principal del proyecto es generar un visor de libro electrónico en formato flipbook por lo que, en primer lugar, se estudiará en qué consiste dicho formato y que pros y contras posee frente a formatos tradicionales.

En una de las primeras búsquedas encontramos que según autores de Wikipedia (2), la palabra flipbook, en castellano folioscopio o filoscopio, se refiere a un libro que contiene una serie de imágenes que cuando se pasan las páginas rápidamente dan la sensación de que el libro está animado, pero este proyecto está enfocado a un flipbook como nuevo formato en una publicación digital.

El formato flipbook ofrece interactividad con el usuario y aporta la sensación de leer un libro o una revista de forma física al simular el paso de páginas.

Según Editoras de TRAMA (3), una de las ventajas de este formato es que se construyen bajo HTML5 así que son compatibles en todos los dispositivos como ordenadores o tabletas sin importar el sistema operativo que utilicen. El contenido de los flipbooks suelen ser imágenes, a diferencia de los formatos de archivos EPUB (Electronic publication) de los libros electrónicos que utilizan texto. Además, al ser un formato online, no depende de tener el contenido guardado en un dispositivo. Finalmente, es remarcable el hecho de que se les puede añadir diferentes funcionalidades como paginación, zoom o notas por páginas

La utilidad del formato flipbook la podemos ver, por ejemplo, en las empresas, que suelen recurrir a este método para elaborar sus catálogos digitales.

## 4.2. Estudio de plataformas existentes

Hoy en día existen diferentes plataformas web que dan solución al problema de generar un libro en formato flipbook, pero cada una de ellas tiene un enfoque diferente. Estudiar y comprobar el funcionamiento de cada una de ellas nos permitirá concretar una idea más clara sobre cómo desarrollar MoonBook, nuestra aplicación.

Para realizar la búsqueda en Google de estas aplicaciones se han introducido las siguientes frases: flipbook reader, aplicación flipbook, generar flipbook a partir de PDF, compartir flipbook y crear flipbook.

Tras ver qué resultados se obtenían se decidió utilizar cinco de esos resultados: Flipsnack, Paperturn, FlipHTML5, YUMPU y FlipBuilder

### 4.2.1. Flipsnack

Flipsnack<sup>1</sup> está especializada en generar revistas en formato flipbook a partir de un archivo con formato PDF (Portable Document Format) o archivos JPG (Joint Photographic Experts Group). Para comenzar a crear una revista, hay que registrarse y una vez que se obtiene una cuenta, hay que especificar la URL por la que se podrá acceder a los libros creados (hay que introducir nombres hasta que alguno de ellos no exista en la plataforma). A continuación, se adjuntan los archivos y cuando se completa la carga se pueden customizar campos como el color de fondo, incorporar sonidos al girar las páginas y opciones sobre si mostrar barra de navegación o el botón de compartir.

Opciones como pantalla completa, tabla de contenidos o búsqueda están restringidos para una versión de pago. En la Figura 2 se puede observar a la izquierda el editor del libro, a la derecha la interfaz para crear la URL y abajo a la derecha la introducción de datos como el título, la descripción y el tipo de visibilidad para terminar de publicar el libro. De forma gratuita la visibilidad del libro es pública.

Según la página de tarifas de Flipsnack (4), los precios oscilan entre 14\$ y 79\$ mensuales en tarifas básicas y otra configurable para empresas que parte de un precio aproximado de 210\$ mensuales. Tiene un total de cinco tarifas en las que se añaden funcionalidades como personalización e imagen de marca, posibilidad de elaborar diseños y utilizar fuentes personalizadas, mayores opciones de privacidad y seguridad (acceso con contraseña al catálogo

---

<sup>1</sup> Flipsnack: <https://www.flipsnack.com/es/>

y compartir catálogo con personas específicas), opciones de organización de contenido como el uso de carpetas y gestión de equipos de trabajo, interactividad con la inserción de videos de plataformas como Youtube<sup>2</sup> o Vimeo<sup>3</sup> o la detección de enlaces, incorporar mecanismos de descarga y de venta de catálogos y por último estadísticas y diferentes tipos de integraciones.

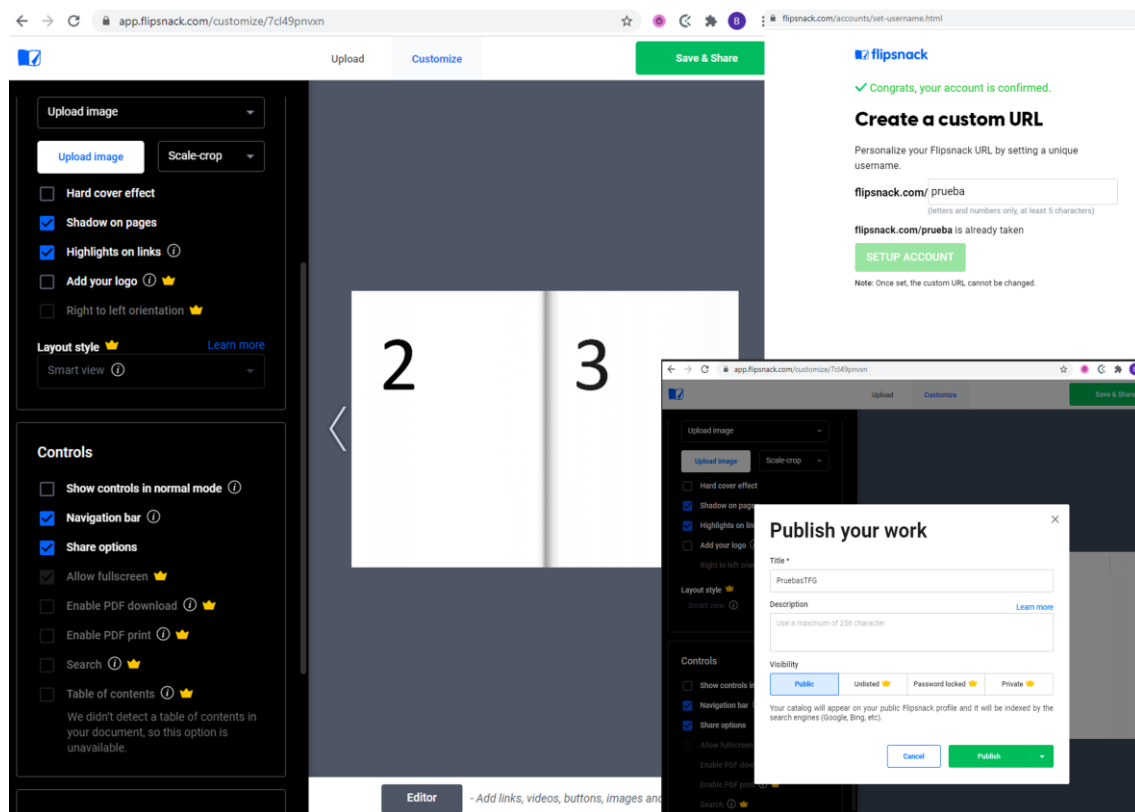


Figura 2 - Varias interfaces de Flipsnack

#### 4.2.2. Paperturn

Al entrar en Paperturn<sup>4</sup>, encontramos en la página principal ejemplos de flipbooks para comprobar cómo funciona el formato y un resumen de las principales características de la aplicación. Para crear una cuenta en la plataforma hay que elegir una de las tarifas disponibles (5), una básica de 11€ y una tarifa pro por 25€ mensuales que incluyen diferentes optimizaciones para buscadores y dispositivos, estadísticas, inserción de videos de distintas plataformas y diferentes opciones de personalización. En Paperturn el precio de la tarifa depende además del número de flipbooks que se quieran crear, los precios anteriores son para la creación y gestión de un solo libro animado.

<sup>2</sup> Youtube: <https://www.youtube.com/>

<sup>3</sup> Vimeo: <https://vimeo.com/es>

<sup>4</sup> Paperturn: <https://www.paperturn.com/es/>

Además de las diferentes tarifas, se encuentra la opción de una prueba gratuita de 14 días (6), para comprobar el funcionamiento interno de la aplicación, se va a elegir esta opción.

Una vez se ha creado una cuenta y tras iniciar sesión, se puede adjuntar el archivo PDF, se añade el título y ya aparece el libro en la pantalla principal de publicaciones. Desde las opciones del libro creado se puede ver la información, editar su contenido e insertar videos, además incluye transformación a código HTML (HyperText Markup Language) para poder insertar el libro en cualquier web.

En la Figura 3, en la parte superior se puede observar el visor, con las diferentes acciones en la parte superior (buscador de texto, visión de todas las páginas en miniatura, impresión y descarga como PDF, opciones de compartir enlace y opciones de accesibilidad) y en la parte inferior derecha la página principal de la aplicación con todas las acciones que se pueden realizar en un libro creado.

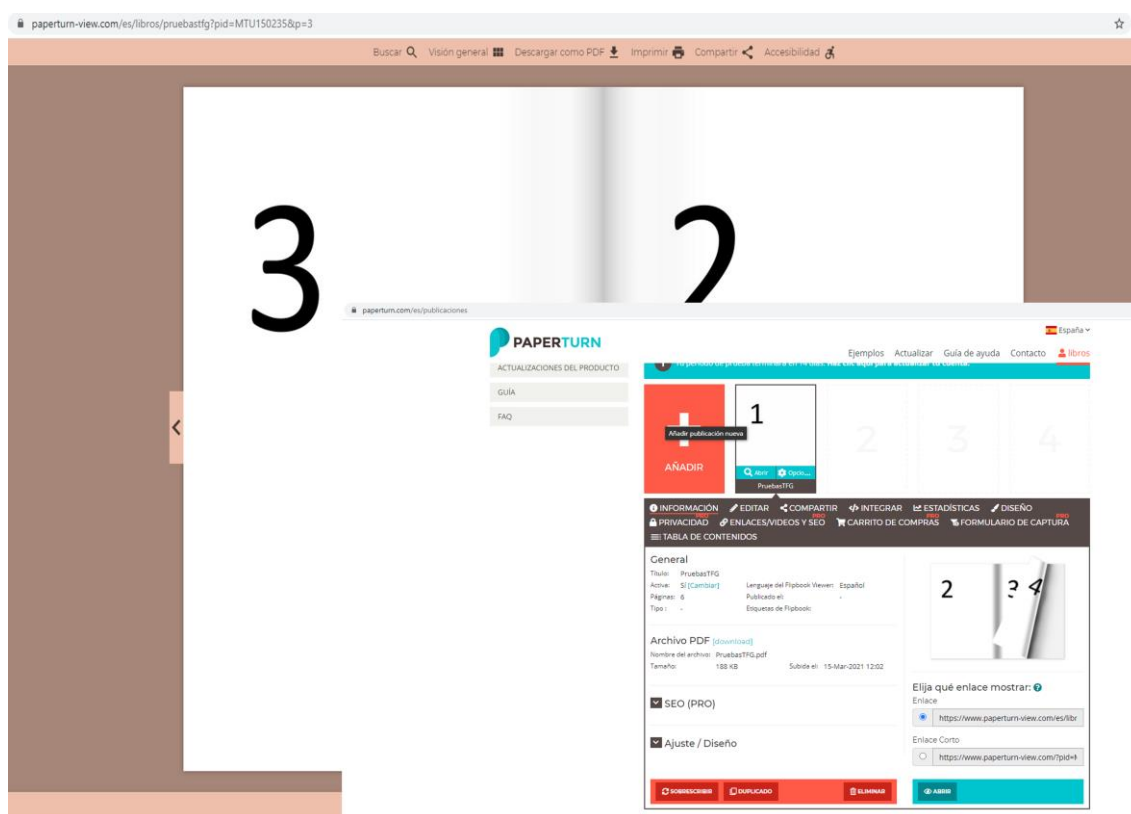


Figura 3 - Varias interfaces de Paperturn

### 4.2.3. Fliphtml5

En la página principal de Fliphtml5<sup>5</sup> se encuentran varios libros como demo para que el usuario pueda ver en qué consiste la herramienta. Para crear un libro es necesario registrarse e iniciar sesión. Una vez dentro, se puede seleccionar la opción de adjuntar ficheros y tras esperar unos segundos nos muestra la visualización del libro como se puede ver en la Figura 4, incluyendo opción de imprimir, pantalla completa, lupa, buscador y opción de compartir a través de un enlace.

En el modo edición nos ofrece la posibilidad de incluir título, descripción, palabras clave, categoría y diferentes estilos de presentación. Además, presenta la opción de poder embeber el visor en un código HTML propio. De forma gratuita solo se puede crear un libro

Al igual que el resto de las plataformas cuentan con una página de tarifas (7), Fliphtml5 tiene un total de cinco tarifas incluyendo la versión gratuita. Los precios oscilan entre los 12,5\$ y 83,5\$ mensuales, incrementando funcionalidades según la tarifa.

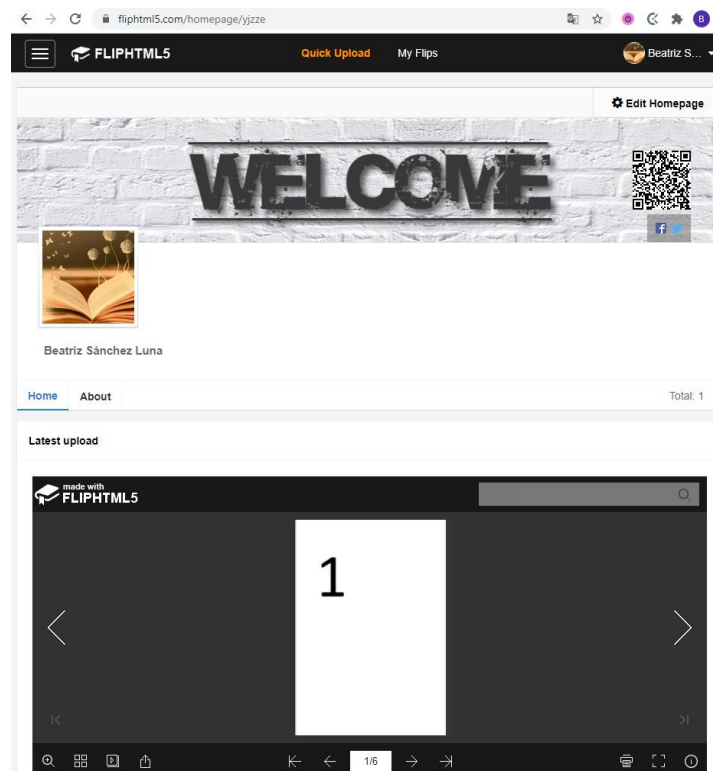


Figura 4 - interfaz FLIPHTML5

<sup>5</sup> Fliphtml5: <https://fliphtml5.com/>



#### 4.2.4. Yumpu

En Yumpu<sup>6</sup> para comenzar a crear un libro hay que tener una cuenta en la plataforma. Una vez dentro de la aplicación, se encuentra en la parte izquierda un menú vertical con la mayoría de las opciones bloqueadas, para su uso hay que tener una tarifa contratada, en la página de precios (8) se puede observar que los precios van desde los 14,95€ a los 178€ mensuales como se puede ver en la Figura 5.

La parte central de la página está dedicada directamente a la subida de archivos y a insertar la información básica del libro a crear. El visor no tiene el efecto de pasar las páginas de un libro físico y se pasan pulsando en unas flechas al igual que un ebook. En el modo edición de libro se pueden modificar campos como título, descripción o idioma, pero no es posible cambiar el contenido del libro una vez creado. En la versión gratuita el libro solo puede ser privado o público para toda la web.

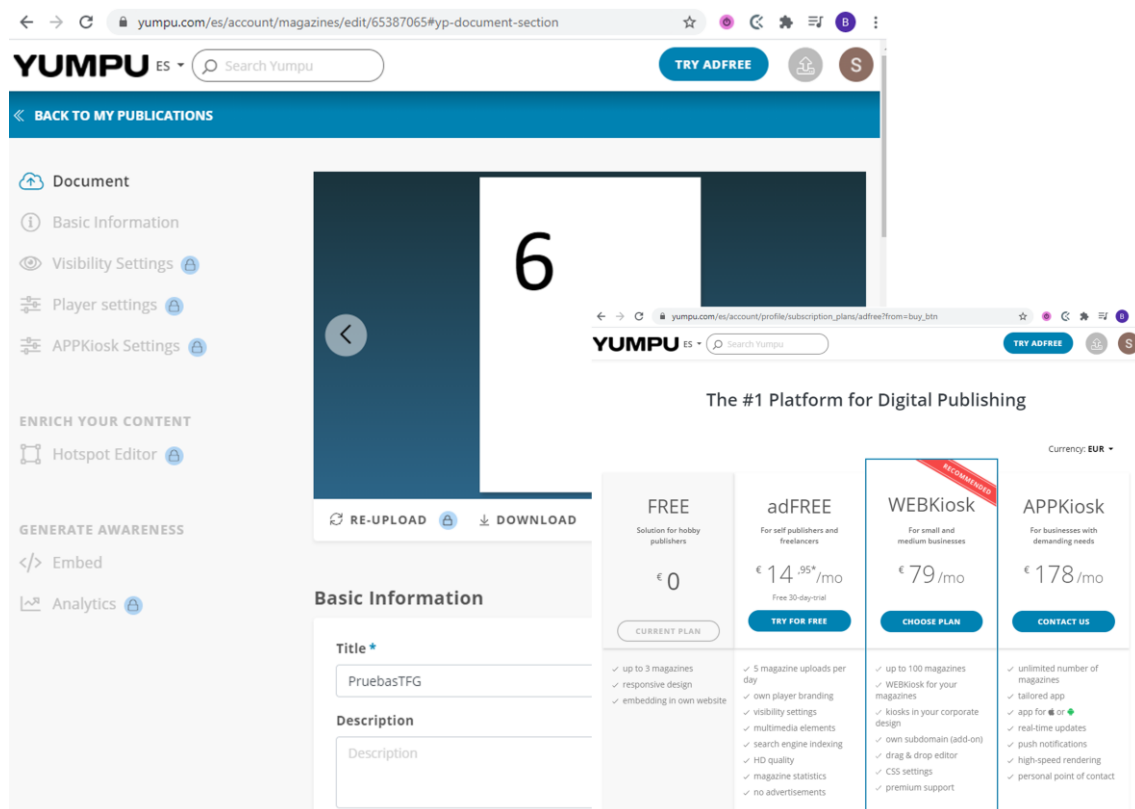


Figura 5 - Varias interfaces YUMPU

<sup>6</sup> Yumpu: <https://www.yumpu.com/es>

## 4.2.5. Flip Builder – Flip PDF Professional

Flip Builder<sup>7</sup> se dedica a realizar software para publicar revistas, folletos o libros interactivos de forma online en formato flipbook. En el apartado productos (9) se encuentran los diferentes software disponibles, uno de ellos es Flip PDF Professional (10), herramienta disponible para Windows, Apple y Android, una vez descargado el programa en el escritorio, permite crear el libro utilizando HTML5 o FLASH, una vez elegida la tecnología se importa un archivo y nos deja controlar la calidad y el tamaño de las páginas o la opción de aplicar una marca de agua. Cuando termina la creación del libro, se muestra el visualizador de forma muy parecida a las aplicaciones anteriores, con la posibilidad de pasar las páginas, aplicar sonido, pantalla completa y zoom. Para realizar la mayoría de las acciones hay que comprar la licencia por un precio de 299\$.

En la Figura 6 se puede observar en la parte inferior izquierda la interfaz de los primeros pasos para crear un libro y en la parte superior de la imagen el visualizador.

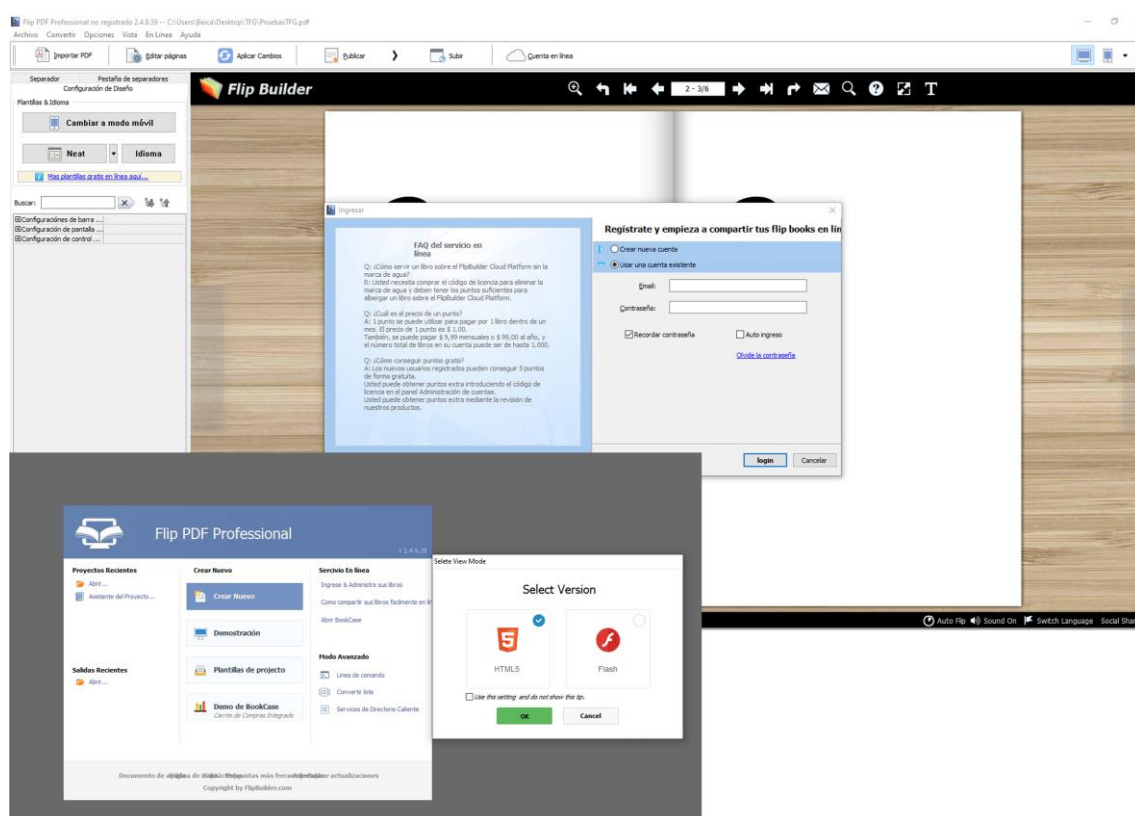


Figura 6 - Varias interfaces de FlipPDF Professional

<sup>7</sup> Flip Builder: <http://flipbuilder.es/>

#### 4.2.6. Conclusiones

Tras utilizar y analizar diferentes plataformas existentes, podemos observar que son todas muy parecidas, la interfaz de visualización casi idénticas, cambiando colores, pero casi todas presentan las mismas funcionalidades básicas como el efecto de pasar de página, zoom y pantalla completa. Otra funcionalidad que se repite es la de compartir, aunque se presenta de diferentes formas: embeber código HTML en web propia, compartir enlace o a usuarios de la misma plataforma, pero en la mayoría de las aplicaciones es una funcionalidad de pago al igual que la inclusión de métricas y estadísticas.

Con respecto a la edición del libro una vez creado, muchas de ellas dejan cambiar títulos y descripciones, pero editar el contenido del libro solo en algunas versiones de pago.

Finalmente, en todas ellas, podemos observar que hay que realizar tres pasos para lograr crear el libro: iniciar sesión, adjuntar archivos e insertar título y descripción. Una vez realizados, es el momento de visualizar el libro.

### 4.3. Librerías existentes para visualizar un flipbook

Tras estudiar algunas de las plataformas que podemos encontrar en el mercado, en este apartado, se estudiará qué librerías existen actualmente que ayuden a la generación de un libro animado.

#### 4.3.1. Turn.js

Turn.js (11) es una librería de JavaScript que convierte el contenido en aspecto de libro utilizando HTML5. Es compatible con los navegadores Safari 5, Chrome 16, Firefox 10 y IE 10,9,8. Posee una documentación de la API (Application Programming Interface) bastante completa y ofrece buenos resultados.

### 4.3.2. OneBook3D

ONEBook3D (12) está disponible como plugin para WordPress<sup>8</sup> y como JQuery<sup>9</sup> plugin. Es compatible con los navegadores más usados y con dispositivos con sistema operativo IOS y Android. Depende de las librerías JQuery, MouseWheel<sup>10</sup> y ThreeJs<sup>11</sup>.

### 4.3.3. BookBlock

BookBlock<sup>12</sup> es un plugin de jquery que permite crear un libro y navegar a través de él pasando las páginas.

### 4.3.4. Comparación

Tras estudiar qué librerías podrían ser útiles para realizar el visor de libros, se han probado y observado el efecto que ofrecen cada una de ellas, en Tabla 2 podemos ver una comparativa con los diferentes aspectos que ofrecen.

*Tabla 2 - Comparativas librerías flipbook*

|   | <b>Turn.js</b>  | <b>OneBook3D</b>  | <b>BookBlock</b>  |
|---|---|---|---|
| Responsive                                      | No  | Si  | Si  |
| Efecto pasar página desde la esquina            | Si  | No  | No  |
| Efecto pasar página desde el borde de la página | No  | Si  | No  |
| Pasar página con botón                          | Si  | Si  | Si  |
| Efecto portada rígida                           | Si  | No  | Si  |
| Barra de progreso lectura                       | Si  | Muestra número de página actual                               | No  |
| Menú secciones libro                            | Si  | No  | No  |
| Imprimir  | No, se imprime la página sin mostrar el contenido del libro | Si, se imprime la pantalla con las páginas del libro visibles | Si, Se imprime la pantalla con las páginas del libro visibles |

<sup>8</sup> Wordpress: <https://es.wordpress.org/>

<sup>9</sup> JQuery: <https://jquery.com/>

<sup>10</sup> MouseWheel: <https://github.com/jquery/jquery-mousewheel>

<sup>11</sup> ThreeJs: <https://threejs.org/>

<sup>12</sup> BookBlock: <https://github.com/codrops/BookBlock>

|                     |    |    |   |
|---------------------|----|----|---|
| Guardar imagen como | No | No | Si, se pueden guardar las páginas del libro haciendo guardar imagen |
|---------------------|----|----|---|

#### 4.3.5. Conclusión

Una vez probadas las diferentes librerías, Turn.js ofrece mejores resultados y es una de las más completas y utilizadas. Esta librería no tiene la opción de imprimir, pero no es un problema, ya que no se ha establecido como un requisito en este proyecto. Para formar el libro utilizan formatos como PNG (Portable Network Graphics) o JPG.

### 4.4. Librerías existentes para transformación de formatos

Además de estudiar librerías que faciliten el desarrollo del visor, es importante como se ha observado en el estudio de plataformas existentes, que es necesario adjuntar archivos para crear el visor. En la mayoría de las aplicaciones estudiadas, para crear el libro, se adjunta un fichero en formato PDF y que, para utilizar las librerías de libros animados, el contenido se muestra en forma de imágenes.

En este apartado se estudiará las librerías de JavaScript existentes para la transformación de formatos de PDF a PNG/JPEG y extraer contenido de un archivo comprimido de extensión ZIP para facilitar al usuario la subida de ficheros.

#### 4.4.1. De PDF a imagen

- **PDF.js<sup>13</sup>**

Librería de JavaScript para analizar y representar archivos PDF. Según Useful Angle (13), una vez que se consigue representar el archivo PDF en una etiqueta <canvas> utilizando esta librería, se puede convertir en imagen utilizando el método `canvas.toDataURL`.

- **Convert-pdf-png<sup>14</sup>**

<sup>13</sup> PDF.js: <https://mozilla.github.io/pdf.js/>

<sup>14</sup> Convert-pdf-png; <https://www.npmjs.com/package/convert-pdf-png>

- **PDFTRON**<sup>15</sup>

#### 4.4.2. Extraer contenido archivo comprimido

- **JSZip**<sup>16</sup>

(14) Librería de JavaScript que permite crear, leer y editar archivos .ZIP.

La función `loadAsync()` de esta librería nos permitirá obtener los ficheros de su interior para su posterior subida al servidor.

---

<sup>15</sup> PDFTron: <https://www.pdftron.com/documentation/samples/js/PDFDrawTest>

<sup>16</sup> JSZip: <https://stuk.github.io/jszip/>

## 5. Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación web que integre un visor de libros electrónico en formato flipbook para que los usuarios puedan crear sus propios libros adjuntando sus archivos y a continuación poder visualizarlos aportando la sensación de leer un libro en papel, al mismo tiempo que pueden compartirlos con sus amigos dentro de la aplicación.

Para conseguir el objetivo principal, podemos definir los siguientes objetivos específicos:

- Realizar un estudio de las plataformas existentes para visualizar libros en formato flipbook.
- Estudiar diferentes librerías que permitan la visualización del libro en HTML y las librerías disponibles para la transformación de archivos.
- Realizar un análisis de las necesidades para obtener el listado de requisitos tanto funcionales como no funcionales.
- Estudiar un diseño adecuado, sencillo e intuitivo.
- Implementar una aplicación web que permita realizar todas las funcionalidades descritas en el análisis y especificación.
- Validar la aplicación con potenciales usuarios de esta aplicación web.

## 6. Metodología

En este apartado se describe la metodología utilizada durante el desarrollo del proyecto, estudiar las diferentes metodologías antes de iniciar un proyecto de desarrollo de software es importante para ver la forma en la que se irá creando la aplicación y las tecnologías a utilizar para la organización de las tareas. El primer paso es elegir entre seguir una metodología tradicional o una metodología ágil.

Según Bryan Molina, Harry Vite y Jefferson Dávila en el artículo de metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software (15), en una metodología de desarrollo tradicional, el proyecto se concibe con una estructura definida, en el que el proyecto se desarrolla de manera secuencial, en una sola dirección y sin posibilidad de volver al estado anterior. Los requerimientos se especifican una vez y para todo el proyecto y no se enseña el resultado al cliente hasta que el proceso ha sido totalmente terminado. En una metodología ágil, el desarrollo es más flexible, se divide el trabajo en tareas más pequeñas, hay una comunicación constante con el usuario obteniendo retroalimentación y es más adaptable a cambios.

Para realizar el proyecto se utilizará una metodología ágil, en concreto la metodología Scrum, ya que es necesaria una retrospectiva constante y tener flexibilidad ante cualquier cambio, además es una metodología para trabajo en equipos pequeños, en este caso formado por dos personas, el tutor y yo, en el que los requerimientos se convierten en funcionalidades del software. Según el artículo anterior, en Scrum se realizan etapas de trabajo denominadas Sprint, donde por cada sprint se realiza una reunión de planificación, se analiza, se diseña, se produce el desarrollo y finalmente se realiza una revisión y retrospectiva.

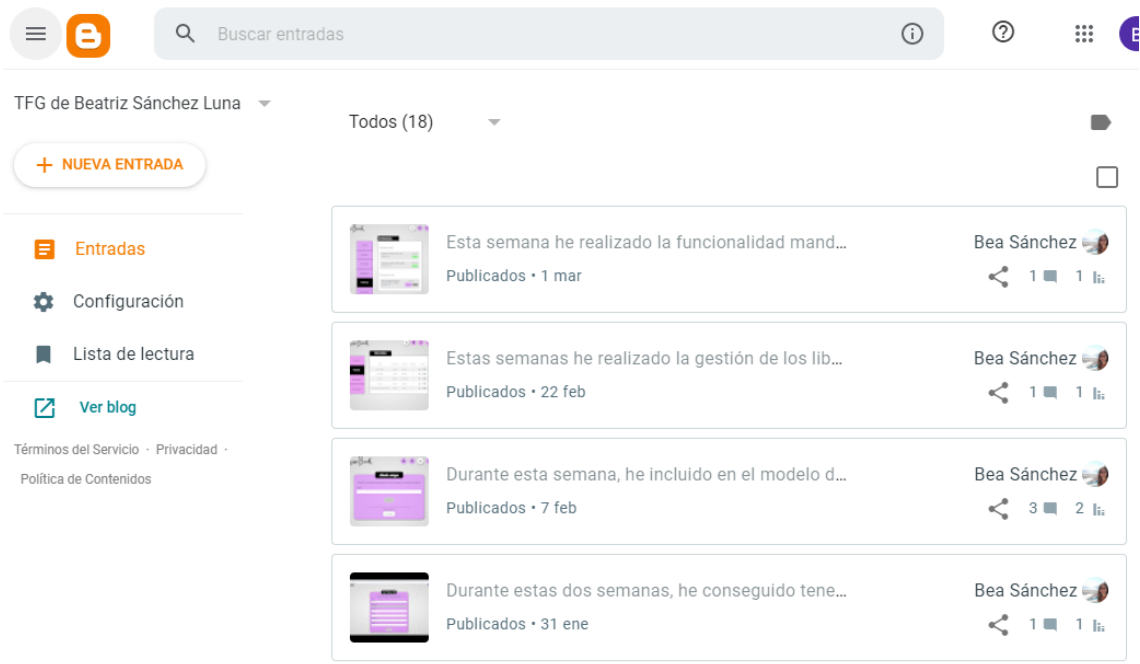
Las iteraciones tendrán una duración de dos semanas, al final de cada iteración se mostrará al tutor el avance del proyecto para ir mejorando aspectos progresivamente.

Para el seguimiento del proyecto se utilizará la plataforma Blogger<sup>17</sup>, la cual permite en un blog privado realizar post semanales y escribir comentarios. Esta plataforma envía un correo de notificación cada vez que se escribe una entrada o un comentario. Este blog quedará de guía para la elaboración de la memoria como se puede ver en la Figura 7.

---

<sup>17</sup> Blogger: <https://www.blogger.com/>





*Figura 7 - Entradas Blog TFG de Beatriz Sánchez Luna*

Para la gestión del proyecto se utilizará Trello<sup>18</sup>, una herramienta que permitirá crear diferentes columnas para organizar las tareas. En este proyecto se han creado 5 columnas: Backlog, dónde estarán todas las tareas pendientes de realizar, en la columna To\_Do, las tareas a realizar por sprint, en la columna In\_Progress, las tareas que están actualmente en progreso, en la columna Waiting se agruparán las tareas que se han quedado en pausa a la espera de preguntar al tutor o de realizar otra tarea y finalmente en la columna Done estarán las tareas terminadas.

En cada iteración se pondrán unas tareas en To\_Do, las cuales al final de la iteración pasarán a la columna Done si se han conseguido realizar. En la Figura 8 podemos ver el estado del tablero a mitad del desarrollo del proyecto.

<sup>18</sup> Trello: <https://trello.com/>

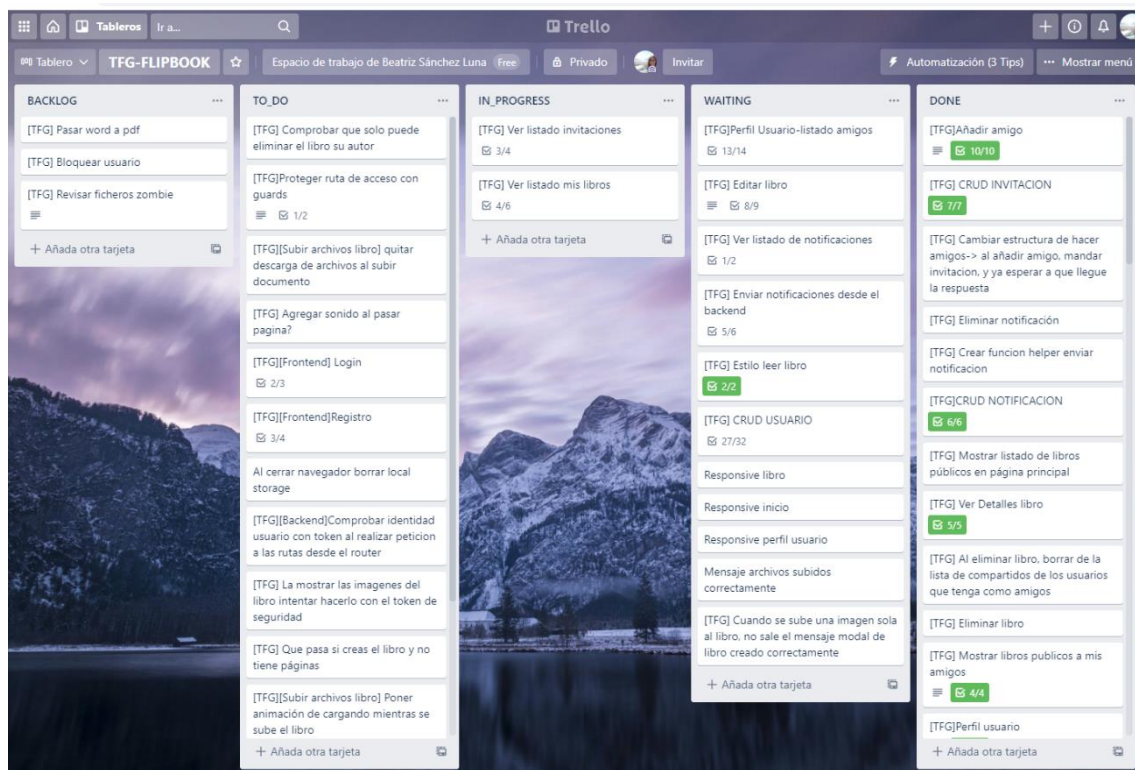


Figura 8 -Tablero TFG en Trello a mitad del desarrollo

Además, para contabilizar el tiempo empleado por tarea se utilizará Toggle<sup>19</sup>, una herramienta muy sencilla de utilizar que permite saber el tiempo dedicado al proyecto y qué tiempos se ha tardado por tarea con tan solo hacer “Play” al empezar a trabajar y “Stop” al finalizar. Además, es compatible con Trello lo que facilita la toma de tiempos de cada una de las tareas del tablero. En la Figura 9 podemos ver una gráfica que nos ofrece Toggle a mitad del desarrollo del proyecto.

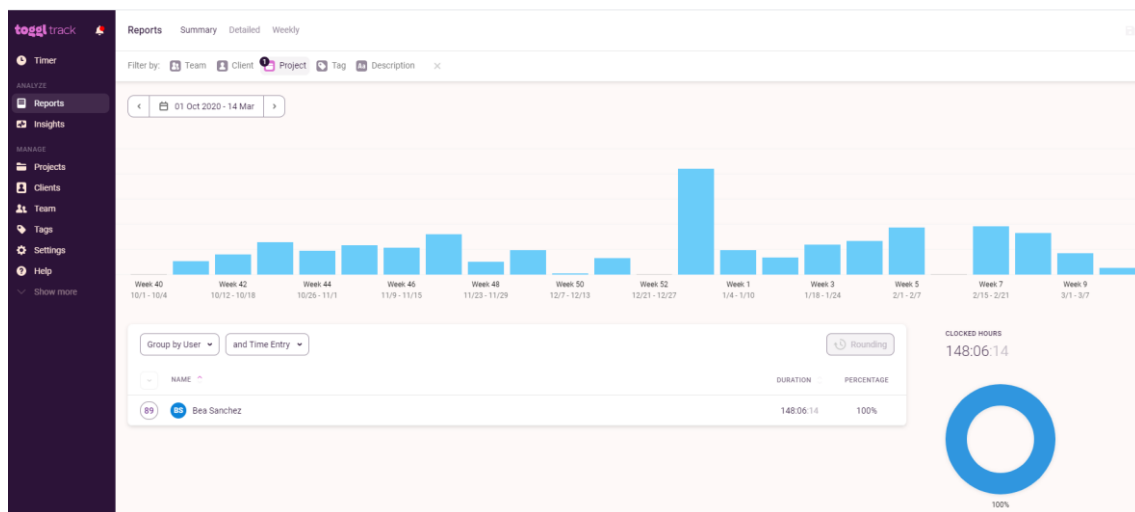


Figura 9 - Gráfico de tiempos en Toggle a mitad de desarrollo

<sup>19</sup> Toggle: <https://toggl.com/>

Finalmente para evitar riesgos durante el desarrollo, el proyecto se alojará en GitHub<sup>20</sup>, lo que permitirá tener un control de versiones Git y evitar la pérdida de código. Para guardar los documentos se utilizará Google Drive<sup>21</sup>. Ambas son unas herramientas muy potentes que ya he utilizado en trabajos anteriores y he obtenido buenos resultados utilizándolas.

---

<sup>20</sup> GitHub: <https://github.com/>

<sup>21</sup> Google Drive: [https://www.google.com/intl/es\\_ALL/drive/](https://www.google.com/intl/es_ALL/drive/)

## 7. Análisis y especificación

En este apartado, se realizará el análisis y especificación del proyecto, para ello se utilizará el estándar IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) 830<sup>22</sup> de la Especificación de Requerimientos del Software. Con este análisis se definen los requisitos funcionales y no funcionales que debe cumplir el sistema para satisfacer las necesidades de los usuarios.

El sistema MoonBook, será una aplicación diseñada para facilitar la conversión de ficheros al formato de libro electrónico y a su vez visualizarlos y compartirlos con usuarios de la misma plataforma. MoonBook contará con dos tipos de roles: usuario y administrador, aunque en este proyecto solo se completará las funcionalidades de un rol usuario.

**-Usuario:** podrá registrarse, iniciar sesión, ver colección de libros creados, ver colección de los libros que han compartido sus amigos, ver colección de libros públicos de la comunidad, crear libros adjuntando ficheros propios, gestionar sus libros (editar y eliminar), leer notificaciones, invitar amigos y visualizar su contenido en formato “flipbook”.

**-Administrador:** personal de la aplicación, tendrá acceso a todo el sistema y será el encargado del buen funcionamiento de la plataforma.

A continuación, se detallan los requisitos funcionales y no funcionales. Para cada uno de ellos se muestra una tabla identificada por un código alfanumérico con la estructura TIPO-NÚMERO, donde el tipo puede ser RF (requerimiento funcional) o RNF (requerimiento no funcional). Además, cada requisito tendrá título, una prioridad (baja, media o alta), según la importancia para el funcionamiento básico de la aplicación, Tipo\_Usuario para indicar que roles utilizan el requisito, y una descripción.

### 7.1. Requisitos funcionales

Los requisitos funcionales son aquellos que definen las funcionalidades que el sistema debe cumplir.

*Tabla 3 - RF-01 Hacer registro*

|                      |                |
|----------------------|----------------|
| <b>Identificador</b> | RF-01          |
| <b>Título</b>        | Hacer registro |

<sup>22</sup> IEE 830: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

|                     |  |
|---------------------|--|
| <b>Tipo_Usuario</b> | Usuario  |
| <b>Prioridad</b>    | Alta   |
| <b>Descripción</b>  | Los usuarios podrán registrarse en la plataforma con su nombre de usuario, email y contraseña. |

*Tabla 4 - RF-02 Iniciar sesión*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-02  |
| <b>Título</b>        | <b>Iniciar sesión</b>  |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Alta   |
| <b>Descripción</b>   | Los usuarios podrán iniciar sesión en la plataforma una vez registrados introduciendo email y contraseña |

*Tabla 5 - RF-03 Cerrar sesión*

|                      |                                   |
|----------------------|-----------------------------------|
| <b>Identificador</b> | RF-03                             |
| <b>Título</b>        | <b>Cerrar sesión</b>              |
| <b>Tipo_Usuario</b>  | Administrador, Usuario            |
| <b>Prioridad</b>     | Alta                              |
| <b>Descripción</b>   | Los usuarios podrán cerrar sesión |

*Tabla 6 - RF-04 Crear libro*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-04  |
| <b>Título</b>        | <b>Crear libro</b>   |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Alta   |
| <b>Descripción</b>   | El usuario podrá crear un libro para ello deberá introducir los campos título, resumen, descripción, categoría, idioma, y elegir el modo de compartir el libro. Además, podrá adjuntar ficheros propios pdf, jpeg o png y .zip con las extensiones anteriores. |

*Tabla 7 - RF-05 Gestionar libro*

|                      |                        |
|----------------------|------------------------|
| <b>Identificador</b> | RF-05                  |
| <b>Título</b>        | <b>Gestionar libro</b> |
| <b>Tipo_Usuario</b>  | Administrador, Usuario |

|                    |  |
|--------------------|--|
| <b>Prioridad</b>   | Alta   |
| <b>Descripción</b> | El usuario podrá editar datos como el título, descripción, resumen, forma de compartir o idioma y eliminar el libro. |

*Tabla 8 - RF-06 Visualizar libro*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-06  |
| <b>Título</b>        | <b>Visualizar libro</b>  |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Alta   |
| <b>Descripción</b>   | El usuario podrá leer el libro en formato “flipbook” simulando el paso de las páginas de un libro en formato físico. |

*Tabla 9 - RF-07 Ver detalles libro*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-07  |
| <b>Título</b>        | <b>Ver detalles libro</b>  |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Media  |
| <b>Descripción</b>   | El usuario podrá ver la información de un libro: portada, título, autor, resumen, descripción, idioma y categoría. |

*Tabla 10 - RF-08 Enviar invitación de amistad*

|                      |   |
|----------------------|---|
| <b>Identificador</b> | RF-08   |
| <b>Título</b>        | <b>Enviar invitación de amistad</b>   |
| <b>Tipo_Usuario</b>  | Administrador, Usuario  |
| <b>Prioridad</b>     | Media   |
| <b>Descripción</b>   | El usuario podrá enviar una petición de amistad a otro usuario de la plataforma introduciendo su email. Al agregar amigos podrá compartir sus libros con ellos. Una vez enviada no podrá volver a enviar otra hasta que esta sea rechazada. |

*Tabla 11 - RF-09 Aceptar/rechazar invitación*

|                      |                                    |
|----------------------|------------------------------------|
| <b>Identificador</b> | RF-09                              |
| <b>Título</b>        | <b>Aceptar/Rechazar invitación</b> |
| <b>Tipo_Usuario</b>  | Administrador, Usuario             |

|                    |  |
|--------------------|--|
| <b>Prioridad</b>   | Media  |
| <b>Descripción</b> | El usuario podrá aceptar o rechazar una petición de amistad enviada por un usuario de la plataforma. Tras aceptarla, podrá ver los libros compartidos por ese usuario. |

*Tabla 12 - RF-10 Ver listado amigos*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-10  |
| <b>Título</b>        | <b>Ver listado amigos</b>  |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Media  |
| <b>Descripción</b>   | El usuario podrá ver el listado de sus amigos, con la opción de eliminarlos. |

*Tabla 13 - RF-11 Eliminar amigo*

|                      |   |
|----------------------|---|
| <b>Identificador</b> | RF-11   |
| <b>Título</b>        | <b>Eliminar amigo</b>                                   |
| <b>Tipo_Usuario</b>  | Administrador, Usuario                                  |
| <b>Prioridad</b>     | Media   |
| <b>Descripción</b>   | El usuario podrá quitar usuarios de su lista de amigos. |

*Tabla 14 - RF-12 Ver listado invitaciones*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-12  |
| <b>Título</b>        | <b>Ver listado invitaciones</b>  |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Media  |
| <b>Descripción</b>   | El usuario podrá ver el listado de invitaciones tanto entrantes como salientes, observando la fecha, el usuario y el estado en el que se encuentra. Desde este listado podrá aceptar o rechazar las invitaciones pendientes. |

*Tabla 15 - RF-13 Ver y editar perfil usuario*

|                      |                                    |
|----------------------|------------------------------------|
| <b>Identificador</b> | RF-13                              |
| <b>Título</b>        | <b>Ver y editar perfil usuario</b> |
| <b>Tipo_Usuario</b>  | Administrador, Usuario             |

|                    |  |
|--------------------|--|
| <b>Prioridad</b>   | Media  |
| <b>Descripción</b> | El usuario podrá ver sus datos y modificarlos. |

*Tabla 16 - RF-14 Ver listado notificaciones*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-14  |
| <b>Título</b>        | <b>Ver listado notificaciones</b>  |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Media  |
| <b>Descripción</b>   | El usuario podrá ver el listado de notificaciones entrantes. Estas notificaciones informarán al usuario de acciones de su interés. |

*Tabla 17 - RF-15 Bloquear/Desbloquear usuarios*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RF-15  |
| <b>Título</b>        | <b>Bloquear/Desbloquear usuarios</b>   |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Baja   |
| <b>Descripción</b>   | El usuario podrá bloquear a otros usuarios para que no le pueda mandar invitaciones. |

*Tabla 18 - RF-16 Ver página principal usuario*

|                      |   |
|----------------------|---|
| <b>Identificador</b> | RF-16   |
| <b>Título</b>        | <b>Ver página principal usuario</b>   |
| <b>Tipo_Usuario</b>  | Administrador, Usuario  |
| <b>Prioridad</b>     | Alta  |
| <b>Descripción</b>   | El usuario podrá ver en la página principal su biblioteca privada, los libros compartidos por sus amigos y los libros públicos de la comunidad con opción de filtrar los libros por título. |

*Tabla 19 - RF-17 Ver página inicio*

|                      |                          |
|----------------------|--------------------------|
| <b>Identificador</b> | RF-17                    |
| <b>Título</b>        | <b>Ver página inicio</b> |
| <b>Tipo_Usuario</b>  | Administrador, Usuario   |
| <b>Prioridad</b>     | Alta                     |



|                    |  |
|--------------------|--|
| <b>Descripción</b> | El usuario verá al entrar a la web un resumen de las funcionalidades que podrá realizar y enlace a iniciar sesión. |
|--------------------|--|

## 7.2. Requisitos no funcionales

Los requisitos no funcionales hacen referencia a características del sistema en particular y a sus restricciones.

*Tabla 20 - RNF-01*

|                      |   |
|----------------------|---|
| <b>Identificador</b> | RNF-01  |
| <b>Tipo_Usuario</b>  | Sistema   |
| <b>Descripción</b>   | El Sistema ofrecerá buena experiencia de usuario. Para ello se realizarán diseños pensando que sean intuitivos y realizar funcionalidades con el menor número de clics. Además se cuidará el contraste de colores para mejorar la accesibilidad y la usabilidad |

*Tabla 21 - RNF-02*

|                      |   |
|----------------------|---|
| <b>Identificador</b> | RNF-02  |
| <b>Tipo_Usuario</b>  | Sistema   |
| <b>Descripción</b>   | El sistema ofrecerá tiempos de respuesta rápidos para mejorar la eficiencia |

*Tabla 22 - RNF-03*

|                      |  |
|----------------------|--|
| <b>Identificador</b> | RNF-03   |
| <b>Tipo_Usuario</b>  | Sistema  |
| <b>Descripción</b>   | El acceso a los datos estará restringido según el tipo de usuario aportando confidencialidad al sistema. |

## 8. Diseño

En este apartado se define el diseño de todo el sistema, desde el diseño de la persistencia de los datos, de la arquitectura, de la API, la arquitectura tecnológica y la interacción con el usuario.

Al final del estudio, se obtienen todos los diseños que servirán como guía para el desarrollo de la aplicación.

### 8.1. Diseño de la persistencia

El diseño de la persistencia hace referencia a todo lo relacionado con el almacenamiento de los datos de la aplicación a desarrollar. En este diseño se decide qué modelo de datos se va a utilizar, por qué tablas o colecciones estará formada y que datos se van a almacenar.

Para construir el modelo de datos podemos optar por utilizar una base de datos relacional o una no relacional. Para realizar una buena elección se estudiará las ventajas y desventajas de cada una de ellas según César Cid Robles (16) y Yecid Alexis Rendón (17):

- **Base de datos relacional:** tienen un identificador que sirve como relación entre un conjunto de datos y otro. La información se almacena en forma de tabla con columnas y filas. Como ventajas encontramos que utiliza lenguaje de consultas SQL (Structured Query Language), presenta mayor soporte y variedad de herramientas como Oracle, MySQL o PostgreSQL entre otras. Además, debe cumplir con los principios ACID (Atomicity, Consistency, Isolation, Durability), por lo que los datos son confiables. Como desventajas encontramos que no es flexible, hay que saber antes de almacenar datos qué datos y cómo estos datos se van a almacenar, requiere mayor procesamiento si la base de datos se hace más compleja. Además, es poco escalable.
- **Base de datos no relacional:** No tienen un identificador que sirva como relación entre un conjunto de datos y otro. Como ventajas encontramos que posee buena escalabilidad, y flexibilidad. Como desventajas encontramos que la integridad de los datos se ve afectada por el poco soporte debido a que es una tecnología relativamente reciente y que en cada base de datos no relacional tiene su propia manera de formatear los datos.

El modelo de datos de MoonBook debe ser flexible y permitir una gran escalabilidad por lo que se usará una base de datos no relacional, en concreto MongoDB.

MongoDB (18) es una base de datos documental, en la que los datos se almacena en forma de documentos tipo JSON (JavaScript Object Notation). Admite matrices y objetos anidados como valores y además trabaja con esquemas dinámicos y flexibles. Aunque la edición de un modelo puede ser más costoso que en una base de datos relacional, las búsquedas son muchas más rápidas.

El primer paso para diseñar el modelo de datos es pensar qué objetos formarán parte de la aplicación: usuarios, libros, invitaciones, notificaciones...

A continuación, se detalla qué colecciones formarán la base de datos y qué datos contendrá:

- **Usuario:** como se puede observar en la Figura 10, un usuario estará formado por un identificador que establece Mongo de manera automática, la fecha de alta, el rol que puede ser usuario o administrador, los campos para almacenar nombre y apellidos, email que será único en la plataforma, un booleano para indicar si el usuario está activo o bloqueado, un campo para guardar la contraseña cifrada, el campo librosCreados que será un array para almacenar los identificadores de los libros creados por el usuario, listaAmigos, un array para almacenar los identificadores de los usuarios que han aceptado la petición de amistad del usuario y listaCompartidos, un array en el que se almacenará los libros que han compartido otros usuarios con él.

El diagrama muestra la estructura de la colección 'Usuario' en MongoDB. Se trata de un documento JSON con los siguientes campos: -Id(Mongoid), -fechaAlta(Date), -rol(string), -nombre(string), -apellidos (string), -email (string), -activo(boolean), -password(string), -librosCreados[(IdLibro)], -listaAmigos:[(IdUsuario)] y -listaCompartidos[(IdLibro)].

```
{
  -Id(Mongoid)
  -fechaAlta(Date)
  -rol(string)
  -nombre(string)
  -apellidos (string)
  -email (string)
  -activo(boolean)
  -password(string)
  -librosCreados[(IdLibro)]
  -listaAmigos:[(IdUsuario)]
  -listaCompartidos[(IdLibro)]
}
```

*Figura 10 - Base de datos colección Usuario*

- **Libro:** como se puede ver en la Figura 11, un libro estará formado por el identificador de Mongo, los campos título, descripción, resumen, fecha de alta, número de páginas

que contiene, el idioma, un booleano para saber si el libro está activo, un campo al que se le llamará compartido que servirá para saber si el libro será público, privado o compartido con amigos, un array de páginas, en el que por cada posición se almacenará el nombre del archivo que se almacenará en el servidor, y la posición que ocupa en el total del libro, también se almacenará un campo categoría y en un campo autor en el que se almacenará el identificador del usuario que ha creado el libro.

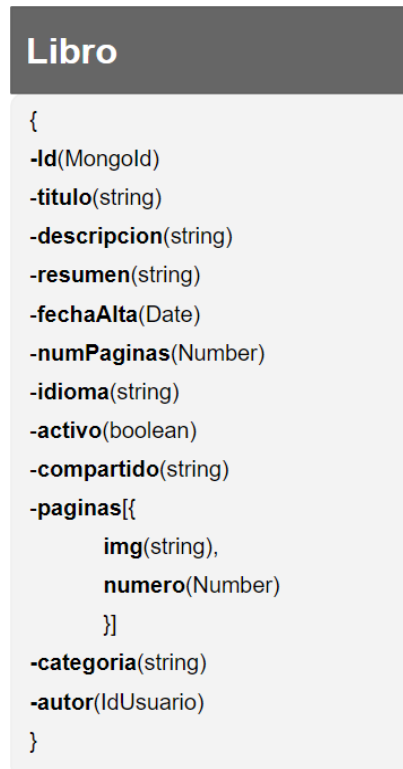


Figura 11 - Base de datos colección Libro

- **Notificación:** como se puede ver en la Figura 12, la colección notificación estará formada por un identificador de Mongo y los campos mensaje y título, además, se guardará en destinatario el identificador del usuario al que va dirigida. Por último, también contendrá los campos fecha de emisión y visto, un booleano para indicar si fuera necesario si una notificación ha sido leída por el usuario o no.

| Notificacion  |
|---|
| <pre>{   -Id(Mongold)   -mensaje(string)   -titulo(string)   -destinatario(Usuariold)   -fechaEmision(Date)   -visto(boolean) }</pre> |

Figura 12 - Base de datos colección Notificación

- **Invitación:** como se puede observar en la Figura 13, la colección invitación está formada por el identificador de mongo, el Id del usuario remitente y destinatario, la fecha de emisión y el estado, que podrá ser pendiente, aceptada o rechazada.

| Invitacion   |
|--|
| <pre>{   -Id(Mongold)   -remitente(Usuariold)   -destinatario(Usuariold)   -fechaEmision(Date)   -estado(string) }</pre> |

Figura 13 - Base de datos colección Invitación

Para mantener la seguridad en la persistencia de los datos, se comprobará diferentes aspectos del usuario para evitar, por ejemplo, que un usuario modifique el contenido de cualquier libro sin ser el creador del mismo.

El primer paso será comprobar que el usuario que está intentando realizar la petición está registrado en la aplicación, en segundo lugar, si hablamos de editar un libro, se comprobará que el creador del libro es el mismo usuario que el que está realizando la petición o tiene rol administrador. De esta forma se asegura que solo el creador del libro o el administrador puede modificarlo o eliminarlo.

## 8.2. Diseño arquitectura conceptual

El diseño de la arquitectura conceptual permite ver a grandes rasgos cómo será la solución que se va a desarrollar. Para la aplicación se utilizará una arquitectura cliente-servidor donde los usuarios realizarán peticiones al servidor para obtener los resultados. El servidor según la petición recibida realizará búsquedas, inserciones o eliminaciones entre otros en la base de datos. Como se puede ver en la Figura 14, el servidor es la única aplicación que puede acceder a los datos, y en función del usuario que esté realizando la petición, devolverá una respuesta u otra a la aplicación del cliente, que se encargará de mostrar los datos recibidos al usuario.

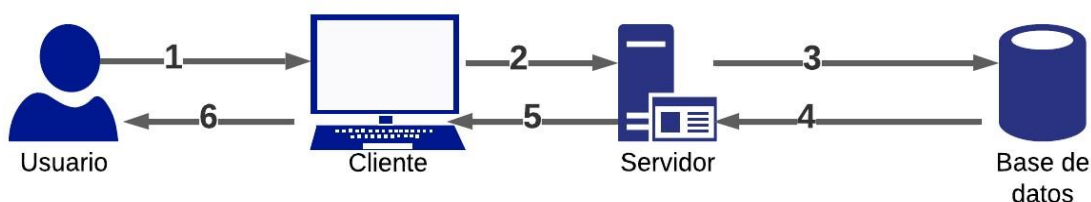


Figura 14 - Esquema arquitectura conceptual realizado con herramienta Lucidchart<sup>23</sup>

## 8.3. Diseño API Rest

Una API debe estar diseñada para que sean sencillas de consumir. Para diseñar una buena API se seguirá el estilo arquitectónico REST (REpresentational State Transfer) siguiendo los principios de diseño de APIS REST de Enrique Amodeo (19): HTTP (Hypertext Transfer Protocol) es un protocolo que se utiliza para realizar los intercambios de información entre el cliente y el servidor. El mensaje de petición en HTTP consta de una primera línea que indica la versión del protocolo, el verbo HTTP (GET, HEAD, PUT, DELETE, POST, OPTIONS) y la URI (Uniform Resource Identifier) de destino, a continuación, se encuentran las cabeceras de manera opcional y, por último, la petición puede llevar un cuerpo que contiene un documento en cualquier formato. Para ello se define el tipo de formato en la cabecera Content-Type. Los mensajes de respuesta siguen la misma estructura excepto la primera línea donde se indica el código de la respuesta que indicará si la petición se ha realizado con éxito o no y un mensaje.

Para el desarrollo de la API se desarrollará un CRUD (Create, Read, Update, Delete) básico para cada una de las colecciones que forman el modelo de datos. A continuación, el listado de rutas

<sup>23</sup> Lucidchart: <https://www.lucidchart.com/>

de la aplicación clasificando las tablas por colores según el tipo de petición: azul (GET, obtener datos), verde (POST, insertar nuevos datos), rojo (DELETE, eliminar datos), naranja (PUT, actualizar datos).

Tabla 23 - API REST GET api/usuarios

| GET         | api/usuarios  |
|-------------|---|
| Descripción | Devuelve todos los usuarios de la plataforma, si se especifica un id, se devuelve un usuario en concreto.   |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (<i>query</i>)</li> <li>• X-token (<i>header</i>) *required</li> </ul>  |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener usuarios<br/>Ejemplo respuesta: <b>Figura 10</b></li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 24 - API REST GET api/usuarios/{id}/libros

| GET         | api/usuarios/{id}/libros  |
|-------------|---|
| Descripción | Devuelve un array con los identificadores de todos los libros creados por el usuario cuyo id es pasado por parámetro  |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (<i>path</i>) *required</li> <li>• Desde (<i>query</i>)</li> <li>• X-token (<i>header</i>) *required</li> </ul>   |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener libros creados por el usuario<br/> <pre>{   "librosCreados":["string"], }</pre> </li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 25 - API REST GET api/usuarios/amigos

| GET         | api/usuarios/amigos   |
|-------------|---|
| Descripción | Devuelve un array con los identificadores de los usuarios que son amigos del usuario que realiza la petición. |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (<i>path</i>) *required</li> </ul>                                |

|                   |  |
|-------------------|--|
|                   | <ul style="list-style-type: none"> <li>• Desde (<i>query</i>)</li> <li>• X-token (<i>header</i>) *required</li> </ul>  |
| <b>Respuestas</b> | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener libros creados por el usuario <pre>{   "listaAmigos":["string"], }</pre> </li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 26 - API REST POST *api/usuarios*

|                    |   |
|--------------------|---|
| <b>POST</b>        | <i>api/usuarios</i>   |
| <b>Descripción</b> | Crear un nuevo usuario  |
| <b>Parámetros</b>  | <ul style="list-style-type: none"> <li>• Usuario (body) *required</li> </ul> <p>Ejemplo modelo:</p> <pre>{   "nombre":"string",   "apellidos":"string",   "email":"string",   "password":"string" }</pre> |
| <b>Respuestas</b>  | <ul style="list-style-type: none"> <li>• <b>200</b> Usuario creado correctamente</li> <li>• <b>400</b> Bad Request</li> </ul>   |

Tabla 27 - API REST POST *api/usuarios/amigo*

|                    |   |
|--------------------|---|
| <b>POST</b>        | <i>api/usuarios/amigo</i>   |
| <b>Descripción</b> | Añadir a lista amigos el nuevo usuario pasado.  |
| <b>Parámetros</b>  | <ul style="list-style-type: none"> <li>• X-token (<i>header</i>) *required</li> <li>• Datos (body) *required</li> </ul> <p>Ejemplo modelo:</p> <pre>{   "amigo":"string", }</pre> |



|            |   |
|------------|---|
| Respuestas | <ul style="list-style-type: none"> <li>• <b>200</b> Usuario creado correctamente</li> <li>• <b>400</b> Bad Request</li> </ul> |
|------------|---|

Tabla 28 - API REST PUT api/usuarios/{id}

|             |  |
|-------------|--|
| <b>PUT</b>  | api/usuarios/{id}  |
| Descripción | Actualizar un usuario ya existente   |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (path): id del usuario a modificar *required</li> <li>• X-token(header) *required</li> <li>• Usuario (body) *required</li> </ul> <p>Ejemplo modelo: <b>Figura 10</b></p> |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Usuario actualizado correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul>   |

Tabla 29 - API REST PUT api/usuarios/amigos/lista

|             |   |
|-------------|---|
| <b>PUT</b>  | api/usuarios/amigos/lista   |
| Descripción | Actualizar lista de amigos de un usuario  |
| Parámetros  | <ul style="list-style-type: none"> <li>• X-token(header) *required</li> <li>• Datos (body) *required</li> </ul> <p>Ejemplo modelo:</p> <pre>{   "amigo": "string", }</pre>      |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Lista amigos actualizada correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 30 - API REST DELETE api/usuarios/{id}

|               |   |
|---------------|---|
| <b>DELETE</b> | api/usuarios/{id}   |
| Descripción   | Eliminar un usuario ya existente  |
| Parámetros    | <ul style="list-style-type: none"> <li>• Id (path): id del usuario a eliminar *required</li> <li>• X-token(header) *required</li> </ul> |

|            |  |
|------------|--|
| Respuestas | <ul style="list-style-type: none"> <li>• <b>200</b> Usuario eliminado correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |
|------------|--|

Tabla 31 - API REST GET api/libros

|             |  |
|-------------|--|
| GET         | api/libros   |
| Descripción | Devuelve todos los libros de la plataforma, si se especifica un id, se devuelve un libro en concreto.  |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (<i>query</i>)</li> <li>• X-token (<i>header</i>) *required</li> </ul>   |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener libros<br/>Ejemplo modelo respuesta: <b>Figura 11</b></li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 32 - API REST GET api/libros/publicos

|             |  |
|-------------|--|
| GET         | api/libros/publicos  |
| Descripción | Devuelve todos los libros públicos de la plataforma  |
| Parámetros  | <ul style="list-style-type: none"> <li>• X-token (<i>header</i>) *required</li> </ul>  |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener libros<br/>Ejemplo modelo respuesta: <b>Figura 11</b></li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 33 - API REST POST api/libros

|             |   |
|-------------|---|
| POST        | api/libros  |
| Descripción | Crear un nuevo libro  |
| Parámetros  | <ul style="list-style-type: none"> <li>• X-token (<i>header</i>) *required</li> <li>• Usuario (<i>body</i>) *required</li> </ul> Ejemplo modelo: <b>Figura 11</b> |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> libro creado correctamente</li> <li>• <b>400</b> Bad Request</li> </ul>                                       |

Tabla 34 - API REST PUT api/libros/{id}

| PUT         | api/libros/{id}  |
|-------------|--|
| Descripción | Actualizar un libro ya existente   |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (path): id del libro a modificar *required</li> <li>• X-token(header) *required</li> <li>• Libro (body) *required</li> </ul> <p>Ejemplo modelo: <b>Figura 11</b></p> |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Libro actualizado correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul>   |

Tabla 35 - API REST DELETE api/libros/{id}

| DELETE      | api/libros/{id}  |
|-------------|--|
| Descripción | Eliminar un libro ya existente   |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (path): id del libro a eliminar*required</li> <li>• X-token(header) *required</li> </ul>                                   |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Libro eliminado correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 36 - API REST GET api/notificaciones

| GET         | api/notificaciones  |
|-------------|---|
| Descripción | Devuelve todas las notificaciones de la plataforma, si se especifica un id, se devuelve una notificación en concreto. Si se especifica un usuario, se devuelven todas las notificaciones de ese usuario   |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (query)</li> <li>• Usuario(query)</li> <li>• X-token (header) *required</li> </ul>  |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener notificaciones</li> </ul> <p>Ejemplo respuesta: <b>Figura 12</b></p> <ul style="list-style-type: none"> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 37 - API REST POST api/notificaciones

| POST        | api/notificaciones  |
|-------------|---|
| Descripción | Crear una nueva notificación  |
| Parámetros  | <ul style="list-style-type: none"> <li>• X-token (header) *required</li> <li>• Notificacion (body) *required</li> </ul> <p>Ejemplo modelo: <b>Figura 12</b></p> |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Notificación creada correctamente</li> <li>• <b>400</b> Bad Request</li> </ul>                              |

Tabla 38 - API REST PUT api/notificaciones/{id}

| PUT         | api/notificaciones/{id}  |
|-------------|--|
| Descripción | Actualizar notificación ya existente   |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (path): id de la notificación a modificar *required</li> <li>• X-token(header) *required</li> <li>• Notificacion (body) *required</li> </ul> <p>Ejemplo modelo: <b>Figura 12</b></p> |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Notificación actualizada correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul>  |

Tabla 39 - API REST DELETE api/notificaciones/{id}

| DELETE      | api/notificaciones/{id}   |
|-------------|---|
| Descripción | Eliminar una notificación ya existente  |
| Parámetros  | <ul style="list-style-type: none"> <li>• Id (path): id de la notificación a eliminar*required</li> <li>• X-token(header) *required</li> </ul>                                 |
| Respuestas  | <ul style="list-style-type: none"> <li>• <b>200</b> Notificación eliminada correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 40 - API REST GET api/invitaciones

| GET         | api/invitaciones  |
|-------------|---|
| Descripción | Devuelve todas las invitaciones de la plataforma, si se especifica un id, se devuelve una invitación en concreto. |

|                   |   |
|-------------------|---|
|                   | Si se especifica un usuario, se devuelven todas las invitaciones de ese usuario   |
| <b>Parámetros</b> | <ul style="list-style-type: none"> <li>• Id (<i>query</i>)</li> <li>• Usuario(<i>query</i>)</li> <li>• Desde (<i>query</i>)</li> <li>• X-token (<i>header</i>) *required</li> </ul>                   |
| <b>Respuestas</b> | <ul style="list-style-type: none"> <li>• <b>200</b> Obtener invitaciones<br/>Ejemplo respuesta: <b>Figura 13</b></li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul> |

Tabla 41 - API REST POST api/invitaciones

|                    |   |
|--------------------|---|
| <b>POST</b>        | api/invitaciones  |
| <b>Descripción</b> | Crear una nueva invitación  |
| <b>Parámetros</b>  | <ul style="list-style-type: none"> <li>• X-token (<i>header</i>) *required</li> <li>• Invitación (body) *required</li> </ul> Ejemplo modelo: <b>Figura 13</b> |
| <b>Respuestas</b>  | <ul style="list-style-type: none"> <li>• <b>200</b> Invitación creada correctamente</li> <li>• <b>400</b> Bad Request</li> </ul>                              |

Tabla 42 - API REST PUT api/invitaciones/{id}

|                    |   |
|--------------------|---|
| <b>PUT</b>         | api/invitaciones/{id}   |
| <b>Descripción</b> | Actualizar invitación ya existente  |
| <b>Parámetros</b>  | <ul style="list-style-type: none"> <li>• Id (path): id de la invitación a modificar *required</li> <li>• X-token(header) *required</li> <li>• Invitación (body) *required</li> </ul> Ejemplo modelo: <b>Figura 13</b> |
| <b>Respuestas</b>  | <ul style="list-style-type: none"> <li>• <b>200</b> Invitación actualizada correctamente</li> <li>• <b>400</b> Bad Request</li> <li>• <b>403</b> Falta de permisos</li> </ul>   |

Tabla 43 - API REST DELETE api/invitaciones/{id}

| DELETE      | api/invitaciones/{id}   |
|-------------|---|
| Descripción | Eliminar una invitación ya existente  |
| Parámetros  | <ul style="list-style-type: none"> <li>Id (path): id de la invitación a eliminar*required</li> <li>X-token(header) *required</li> </ul>                               |
| Respuestas  | <ul style="list-style-type: none"> <li><b>200</b> Invitación eliminada correctamente</li> <li><b>400</b> Bad Request</li> <li><b>403</b> Falta de permisos</li> </ul> |

Tabla 44 - API REST POST api/login

| POST        | api/login   |
|-------------|---|
| Descripción | Iniciar sesión en MoonBook  |
| Parámetros  | <ul style="list-style-type: none"> <li>datos (body) *required</li> </ul> <p>Ejemplo modelo:</p> <pre>{   "email": "string",   "password": "string", }</pre> |
| Respuestas  | <ul style="list-style-type: none"> <li><b>200</b> Login realizado correctamente</li> <li><b>400</b> Bad Request</li> </ul>                                  |

Tabla 45 - API REST GET api/subir/{tipo}/{nombreArchivo}

| GET         | api/subir/{tipo}/{nombreArchivo}  |
|-------------|---|
| Descripción | Devuelve el recurso solicitado  |
| Parámetros  | <ul style="list-style-type: none"> <li>X-token (header) *required</li> </ul>  |
| Respuestas  | <ul style="list-style-type: none"> <li><b>200</b> Recurso devuelto correctamente</li> <li><b>400</b> Bad Request</li> <li><b>403</b> Falta de permisos</li> </ul> |

Tabla 46 - API REST POST api/subir/{tipo}/{id}

| POST | api/subir/{tipo}/{id} |
|------|-----------------------|
|------|-----------------------|

|                    |   |
|--------------------|---|
| <b>Descripción</b> | Subir recurso al servidor pasándole el tipo de recurso y el identificador del libro al que se añade esa imagen.                   |
| <b>Parámetros</b>  | <ul style="list-style-type: none"> <li>• datos (body) type file *required</li> <li>• X-token (<i>header</i>) *required</li> </ul> |
| <b>Respuestas</b>  | <ul style="list-style-type: none"> <li>• <b>200</b> Recurso subido correctamente</li> <li>• <b>400</b> Bad Request</li> </ul>     |

#### 8.4. Diseño arquitectura tecnológica Front/Back-end

En este apartado se especifican que tecnologías se van a utilizar en cada bloque de la aplicación definidos en el apartado 8.2 de la arquitectura conceptual.

Para el desarrollo de la aplicación web se utilizará el paquete tecnológico MEAN, que utiliza el lenguaje de programación JavaScript y agrupa las siguientes tecnologías: MongoDB (base de datos), Express, AngularJs (cliente web) y NodeJs (servidor).

- **MongoDB:** base de datos no relacional como se ha descrito en el apartado 8.1.
- **Express:** (20) framework de Node que ofrece diferentes características como el manejo de rutas, manejo de errores o la integración con la base de datos. Express permitirá crear una API rápida y sencilla.
- **Angular:** (21) framework de código abierto, desarrollado por Google, que facilita la programación de web de una sola página (Web SPA, Single Page Application) utilizando principalmente lenguaje TypeScript. Al utilizar Angular permitirá que la aplicación escale de forma sencilla al estar basada en componentes.
- **Node:** (22) entorno de desarrollo en tiempo de ejecución que se encuentra en la capa del servidor. Utiliza JavaScript como lenguaje de programación, es multiplataforma y está orientado a eventos asíncronos

En la Figura 15 se observa el diagrama definido en la arquitectura conceptual con las tecnologías que se utilizarán.

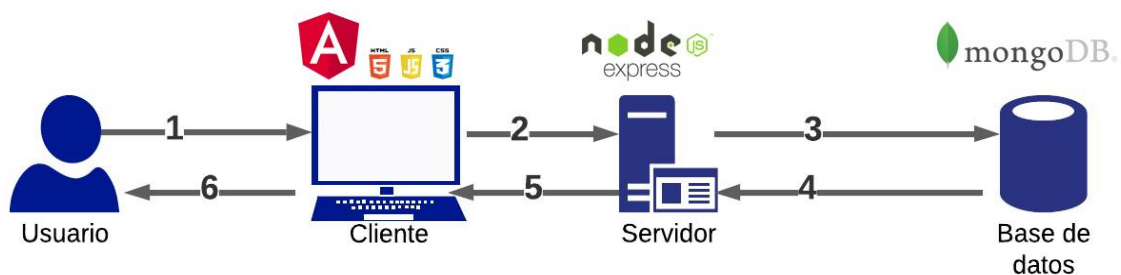


Figura 15 - Arquitectura tecnológica

## 8.5. Diseño interacción o experiencia de usuario

Los usuarios interactuarán con la aplicación a través de una interfaz, el diseño de la experiencia de usuario consiste en hacer que este punto de conexión entre el usuario y la aplicación sea lo más sencilla posible, útil y amigable. Para ello se seguirán algunos de los principios y métodos de la experiencia de usuario escrito por Yuseg Hassan Montero (23) y se detallará la forma en la que se resolverá en nuestra aplicación MoonBook:

- **Colores**

Se utilizará un máximo de cinco colores. Además, se evitará el uso exclusivo del color para realizar una acción u otra y se evitará la combinación de colores incompatibles.

- **Eficiencia – Completar una tarea en el menor tiempo posible**

Para que el usuario logre completar cada tarea en el menor tiempo posible, se procurará que para realizar una acción no tenga que realizar más de 3 clics, los formularios serán lo más breve posible y claros, poniendo etiquetas y ejemplos de qué dato va en cada campo. Además, en los formularios en los que hay campos para elegir entre varias opciones, por ejemplo, al crear un libro, habrá que especificar de qué modo se va a compartir (privado, público, con amigos), al marcar por defecto una de las opciones más utilizadas podemos ahorrar tiempo a muchos usuarios.

- **Evitar errores**

Para evitar que el usuario se equivoque al utilizar la aplicación se limitarán las posibilidades en los formularios, por ejemplo, si debe introducir un número, se limitará la introducción de letras. Si hay algún campo que puede llegar a ser confuso, se utilizarán mensajes de ayuda para reducir la posibilidad de error. Para la realización de acciones que puedan ser irreversibles, como la eliminación de un libro, se avisará al usuario de las consecuencias que tendrá esa acción, y se le dará opción a retroceder si no la quiere



realizar. En el caso que se produzca el error, se informará de forma clara y sencilla de lo sucedido al usuario.

- **Iconos**

Los iconos aportan visualidad y ayudan a interpretar acciones siempre que no sean simplemente decorativos que pueden hacer compleja la interfaz. Se utilizarán iconos para representar acciones como eliminar, editar o volver atrás, en las diferentes acciones del menú de navegación y para distinguir entre los diferentes estados en los que se puede encontrar el campo compartido de un libro (privado, público o con amigos).

- **Jerarquía visual**

La jerarquía visual establece que se debe dar más importancia a los elementos que sean más relevantes para el usuario en ese momento. Por ejemplo, se destacará en todo momento el sitio de la página en el que se encuentra el usuario, cambiando el color o el tamaño.

Todos estos aspectos se tendrán en cuenta para el diseño de las interfaces y en el desarrollo de la aplicación.

## 8.6. Diseño de interfaces

El diseño de interfaces se ha realizado a mano utilizando la aplicación Notas<sup>24</sup> de una tableta iPad Air 3 y el Apple Pencil. Para modificar algunos detalles de los bocetos ya terminados, se ha utilizado la herramienta Adobe Photoshop<sup>25</sup>. A continuación, se mostrarán los diferentes bocetos de MoonBook junto a una pequeña explicación:

El mockup de “página inicio” es la interfaz con la que se encuentra el usuario por primera vez al entrar en MoonBook. La idea es que de un vistazo el usuario pueda ver de forma resumida los principales pasos a realizar para utilizar la plataforma. En primer lugar se realizaron los mockups correspondientes a la Figura 16, Figura 17, Figura 18 y finalmente en la Figura 19, se encuentra el diseño final para la página de inicio que corresponde con el requisito de usuario Tabla 19 - RF-1.

---

<sup>24</sup> Notas: <https://apps.apple.com/es/app/notas/id1110145109>

<sup>25</sup> Adobe Photoshop: <https://www.photoshop.com/en>

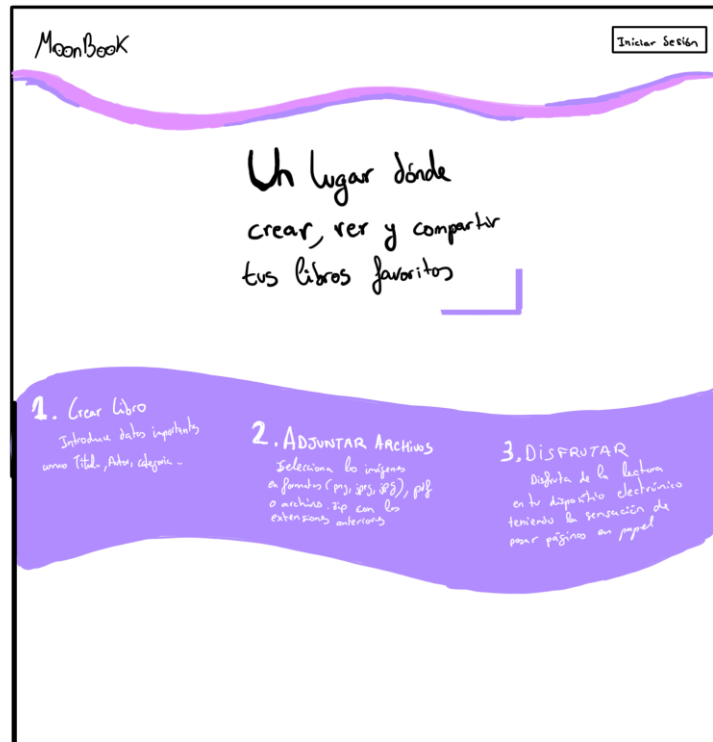


Figura 16 - Mockup 1 Página Inicio

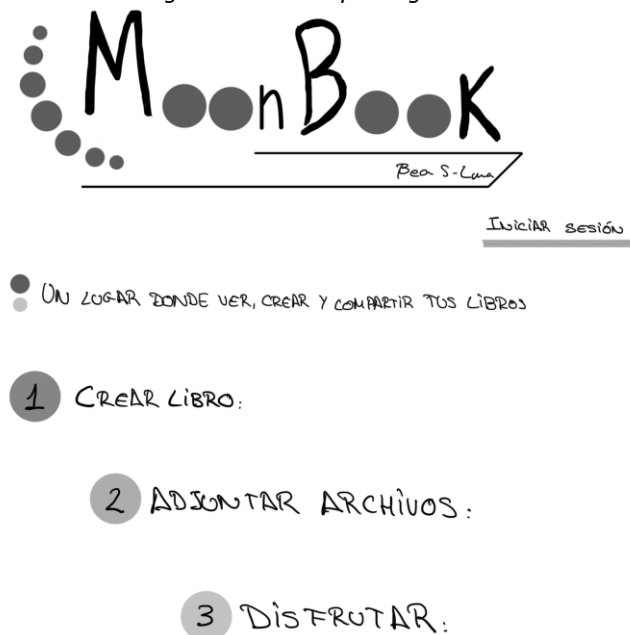


Figura 17 - Mockup 2 Página Inicio



Figura 18 - Mockup 3 Página Inicio

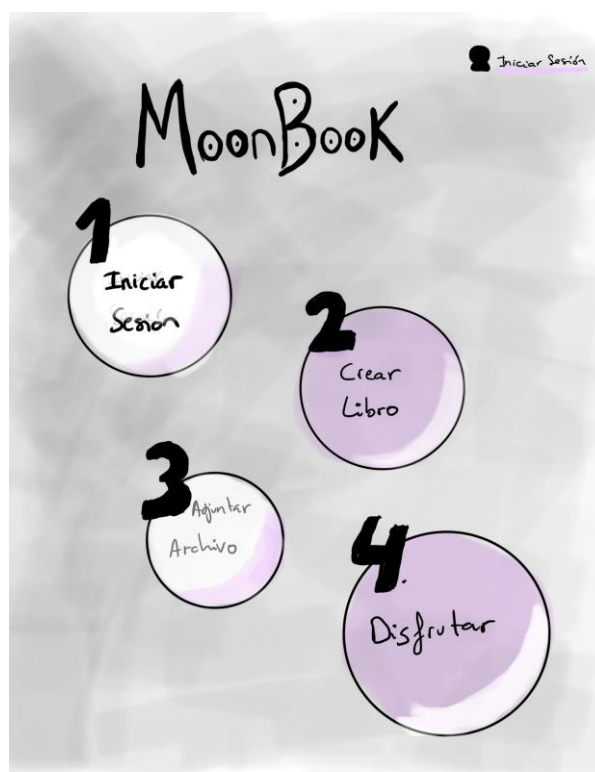


Figura 19 - Mockup final Página Inicio

El mockup de “Registro” de la Figura 20 corresponde al requisito Tabla 3 - RF-01, este formulario permitirá al usuario introducir sus datos por primera vez para a continuación, poder iniciar sesión. El formulario estará centrado en la página y tendrá el botón de enviar los datos y el de Iniciar Sesión.

MoonBook

Crear Cuenta

Nombre

Apellidos

Email

Contraseña

ENVIAR

-----

¿Ya tienes una cuenta?

Iniciar Sesión

Figura 20 - Mockup Registro

El mockup de “Iniciar Sesión” de la Figura 21 corresponde a Tabla 4 - RF-02, el formulario de iniciar sesión sigue el mismo estilo que el de registro, cambiando el botón de ir a iniciar sesión por el de ir a registro. El logo de MoonBook será en ambos formularios un enlace a la página principal (Figura 19 - Mockup final Página Figura 19).



Figura 21 - Mockup Iniciar sesión

En el mockup “Página Principal usuario” correspondiente a la Figura 22 corresponde al requisito de usuario Tabla 18 - RF-16, en esta interfaz el usuario encontrará por cada una de las filas un tipo de libro: privados, públicos o los que han compartido sus amigos. Dentro de estas filas aparecerán los libros, formados por la portada y el título. Al pulsar la imagen de la portada, el usuario será redirigido a visualizar el libro. Además, en la parte derecha de cada miniatura del libro se encontrará el botón para acceder a la información más detallada de ese libro.

En la parte izquierda superior de la pantalla siempre se encontrará el logo el cual al pulsar sobre él nos llevará a esta interfaz de la página principal del usuario.

En la parte derecha superior se encontrará el menú con las principales acciones: Mi Perfil, Añadir Libro, Añadir Amigo y Cerrar Sesión correspondiente al requisito Tabla 5 - RF-03.

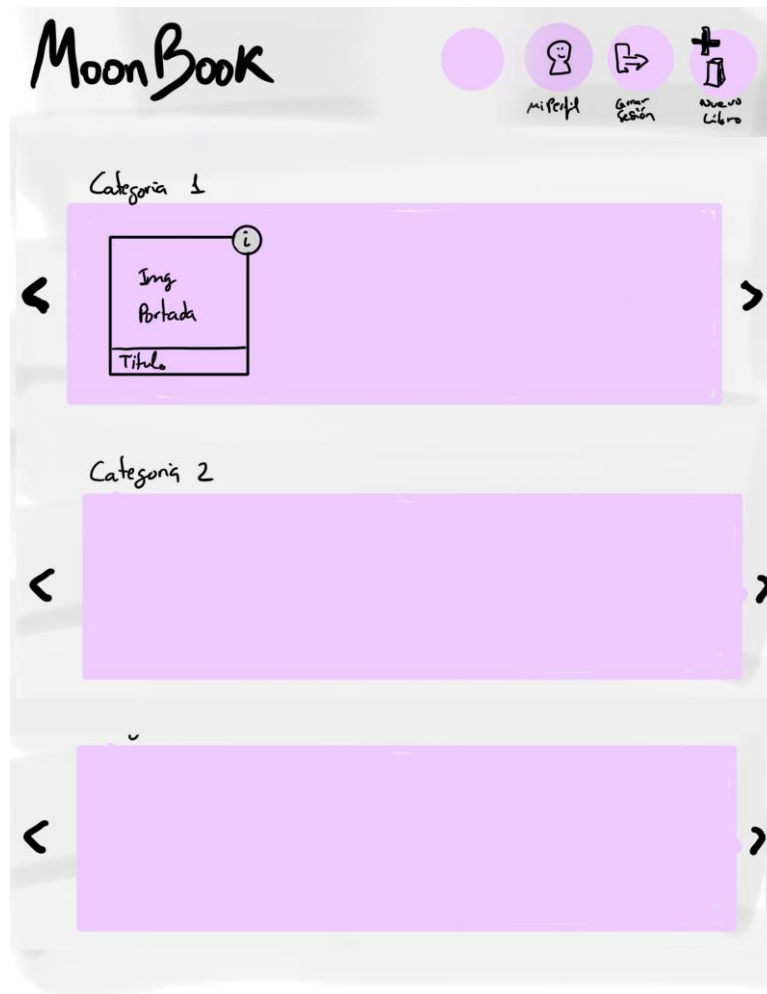


Figura 22 - Mockup Página principal usuario

En el mockup “Perfil usuario” satisface el requisito Tabla 15 - RF-13 se accede desde el menú situado arriba a la derecha, al entrar al perfil, encontraremos un menú secundario a la izquierda, en este menú aparecerán opciones como ver/editar perfil usuario, pestaña que se abrirá por defecto al seleccionar perfil desde el menú superior, ver listado de libros creados y poder gestionarlos, ver notificaciones, ver invitaciones y el listado de amigos.

En el centro encontraremos el contenido que cambiará según en la opción del menú de la izquierda en la que se encuentre el usuario. En la Figura 23, se encontraría seleccionada la opción de ver los datos del perfil del usuario, en la parte derecha del contenedor se encontrará un botón con el icono de un lápiz para ir a editar los datos del perfil, esta interfaz será igual que la actual pero indicando al usuario que los campos se pueden editar y aparecerá al final de los campos del formulario un botón para guardar los cambios realizados y en la parte izquierda del contenedor una flecha para salir del modo edición.

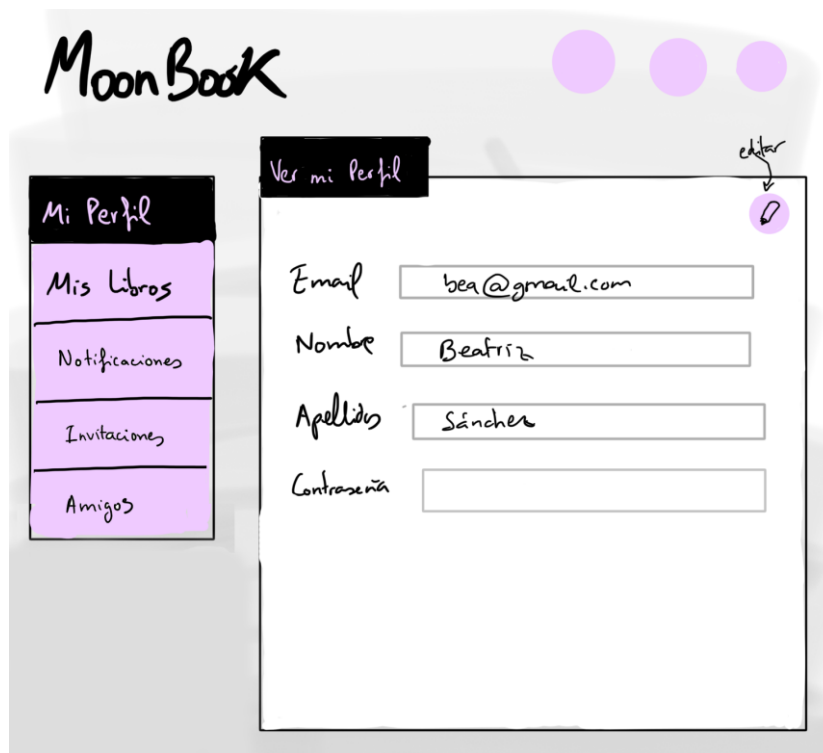


Figura 23 - Mockup Perfil usuario: Ver mi perfil

El mockup “Gestionar libros” de la Figura 24 corresponde al requisito de usuario Tabla 7 - RF-05, en esta interfaz el usuario podrá ver el listado de sus libros creados, por cada una de las filas de la tabla podrá acceder pulsando en el icono del ojo a visualizarlo, en el icono de la papelera podrá eliminarlo, mostrando un mensaje modal para informar al usuario que la acción es irreversible y en el icono del lápiz podrá editarlo.

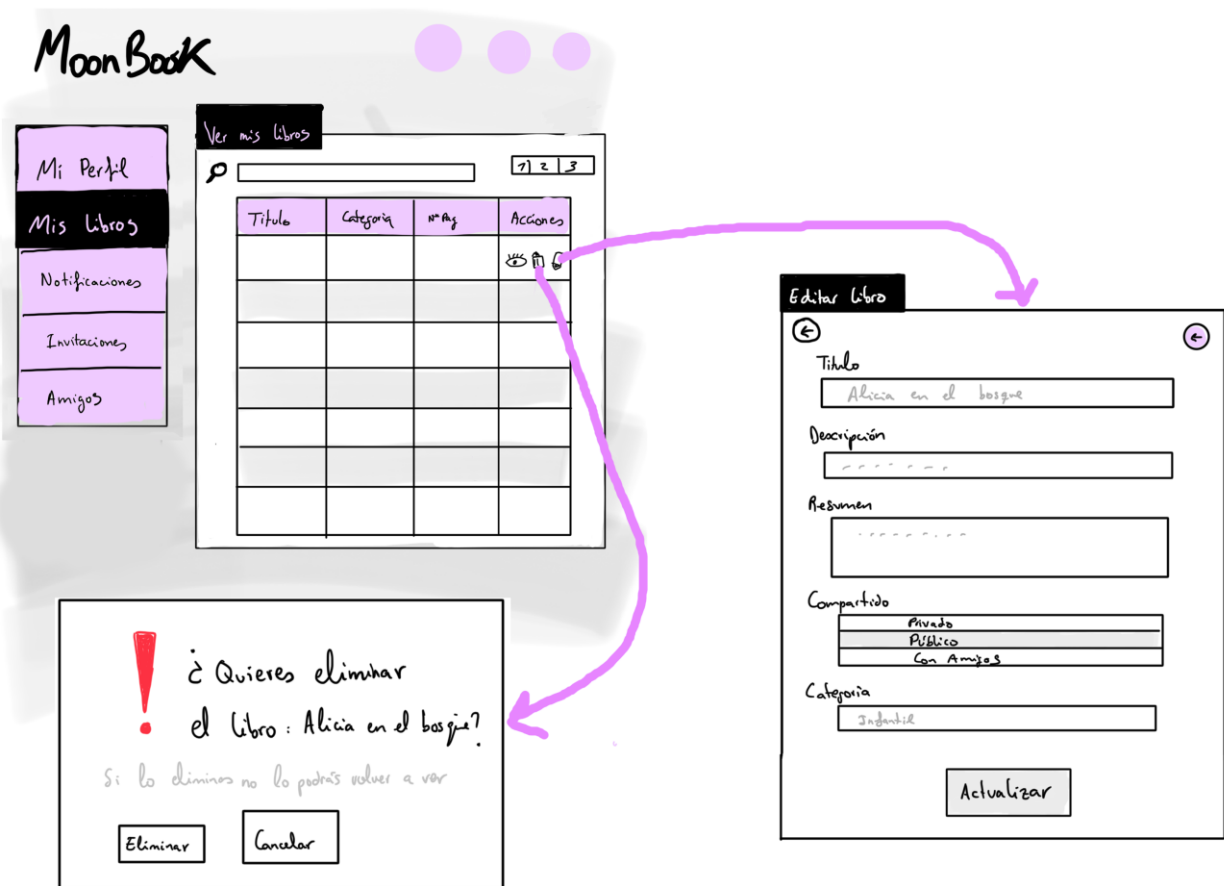


Figura 24 - Mockup Perfil usuario: gestionar libros

En el mockup “notificaciones” de la Figura 25 - Mockup Perfil usuario: notificaciones el usuario podrá ver todas las notificaciones que le llegarán para informar sobre acciones realizadas en la aplicación o para informar sobre los nuevos libros que comparten sus amigos. Para cada una de las notificaciones aparecerá título, descripción, fecha y en la parte derecha del contenedor de la notificación una cruz que permitirá al usuario eliminar la notificación, satisfaciendo el requisito Tabla 16 - RF-14. En este caso no se advertirá al usuario de la eliminación, puesto que no es una acción que implique una gran pérdida al usuario como si lo es por ejemplo borrar un libro.



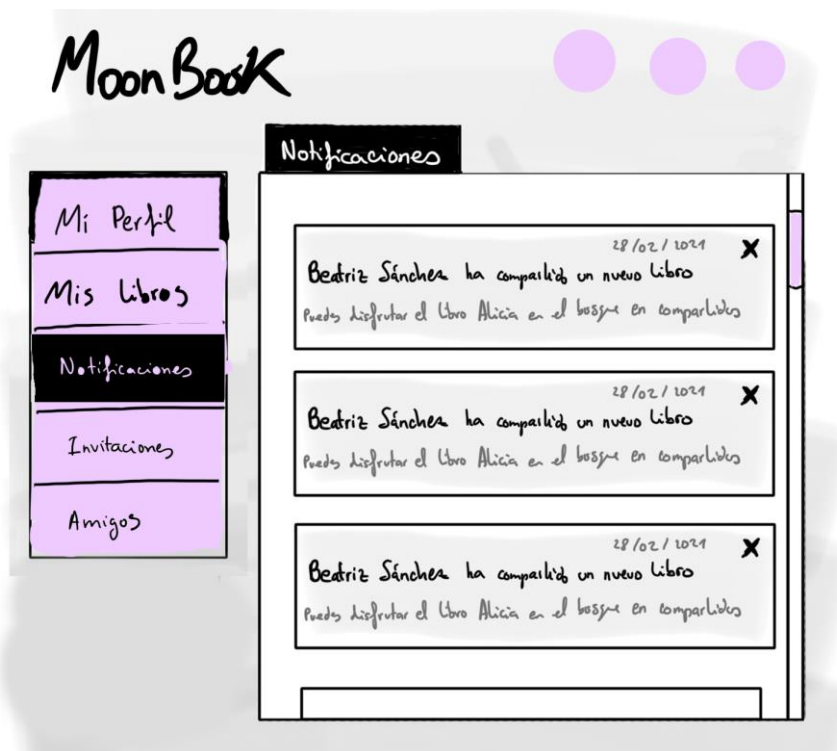


Figura 25 - Mockup Perfil usuario: notificaciones

En el mockup “invitaciones” de la Figura 26 corresponde a los requisitos Tabla 11 - RF-09 y Tabla 14 - RF-12, sigue el mismo estilo que todas las interfaces del perfil del usuario, en este caso, encontraremos tanto el listado de invitaciones que el usuario ha enviado como las que ha recibido, pudiendo aceptarlas o rechazarlas.



Figura 26 - Mockup Perfil usuario: invitaciones

El mockup “amigos” de la Figura 27 - Mockup Perfil usuario: amigosFigura 27 corresponde a los requisitos Tabla 12 - RF-10 y Tabla 13 - RF-11. Siguiendo la misma estructura que en las anteriores interfaces, aparecerá un listado con los diferentes usuarios que son amigos con la opción de eliminarlos. En este caso al eliminar aparecerá un mensaje modal para advertir al usuario que si lo elimina su amigo dejará de ver los libros que anteriormente compartía.

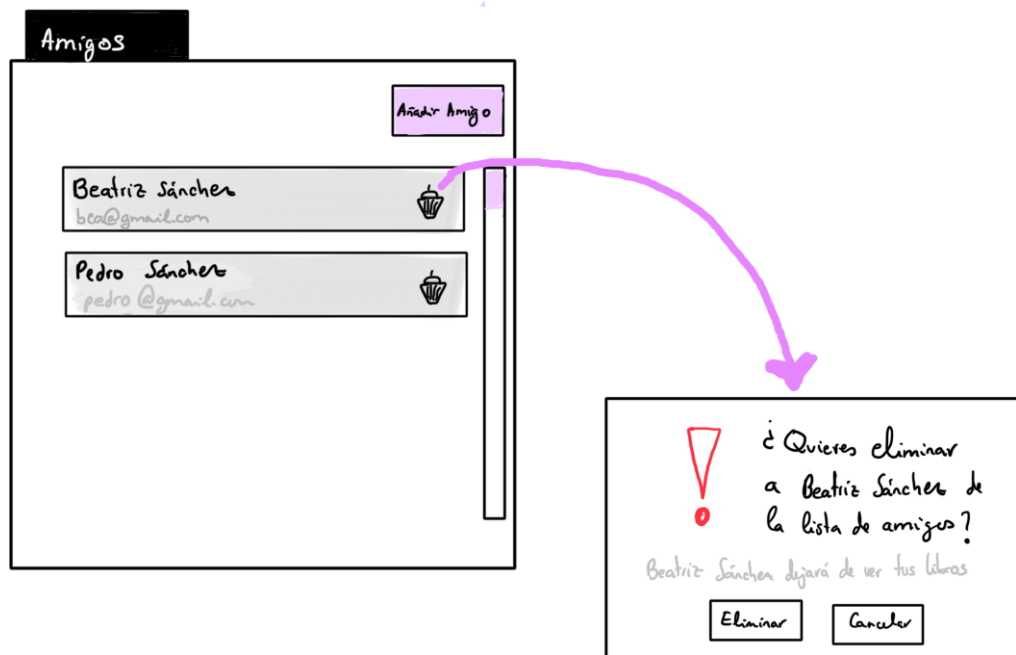


Figura 27 - Mockup Perfil usuario: amigos

El mockup “añadir amigo” de la Figura 28 satisface el requisito Tabla 10 - RF-08, esta interfaz sigue el mismo estilo que en iniciar sesión y registro, en este formulario, el usuario podrá introducir el email de otro usuario y pulsar enviar para que se envíe la invitación. En la parte inferior, se informa de los amigos actuales y un botón para ir a Perfil/amigos. Si existe el email introducido aparecerá un mensaje modal durante unos segundos, en caso contrario se informará del error.

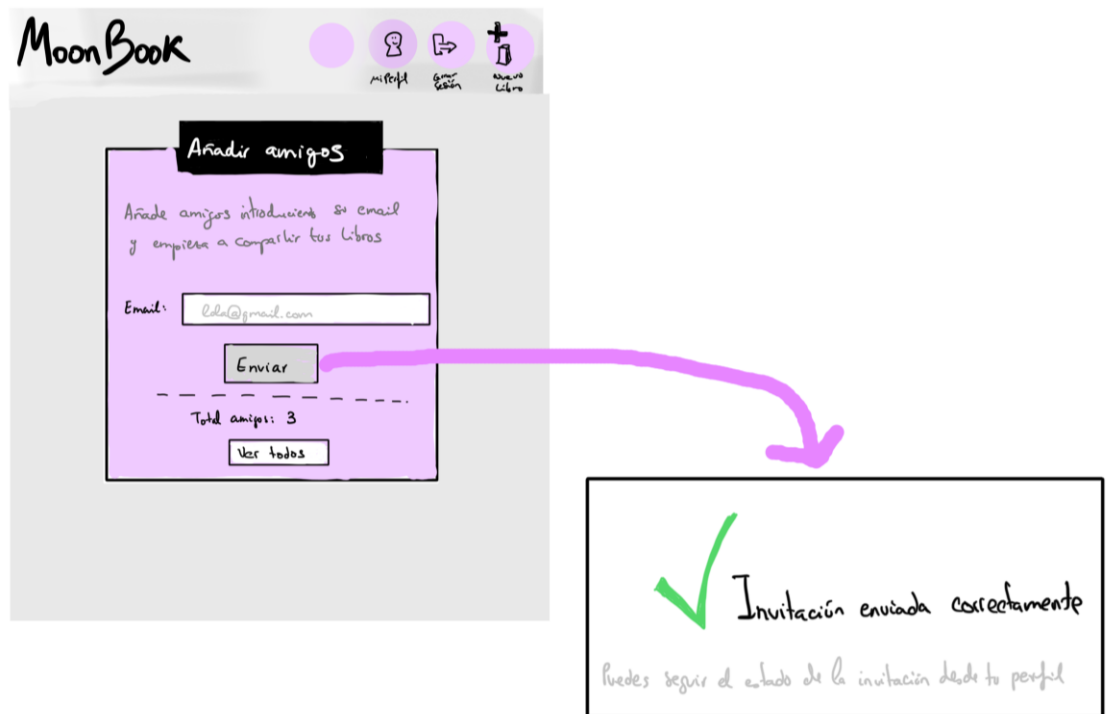


Figura 28 - Mockup añadir amigo

Los mockups “Crear libro” de la Figura 29 satisface el requisito de usuario Tabla 6 - RF-04. Sigue el mismo estilo que los anteriores formularios, con el contenedor centrado en el centro de la pantalla. El formulario se divide en dos pasos: en primer lugar, introducir los datos principales del libro como son título, descripción, resumen, categoría, forma de compartir y el idioma. Y a continuación, tras pulsar el botón siguiente, el usuario pasará a la segunda parte del formulario donde introducirá los archivos que formarán el libro. Tras completar la subida, aparecerá un mensaje modal indicando al usuario el resultado de la acción, dando la posibilidad de volver a la página principal del usuario o ir a visualizar el libro creado. Figura 29 - Mockups añadir libro

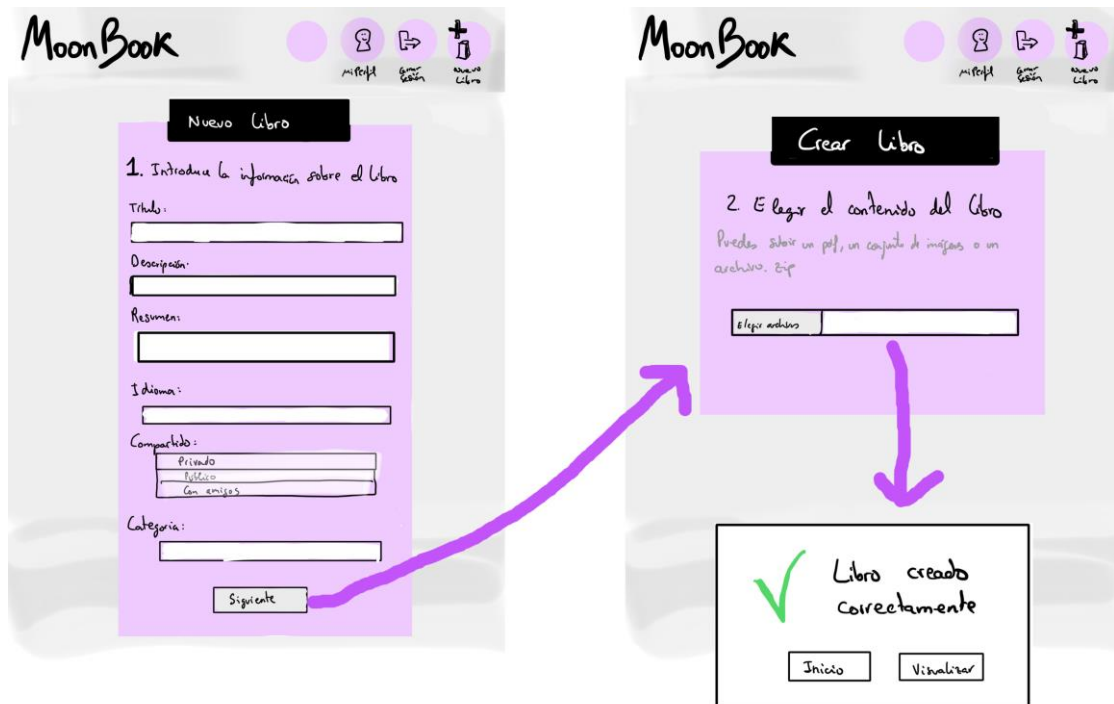


Figura 29 - Mockups añadir libro

El mockup “Visualizar libro” de la Figura 30 corresponde al requisito de usuario Tabla 8 - RF-06, el libro ocupará toda la pantalla, y en la parte izquierda se situarán las acciones que se podrán realizar como el zoom o pantalla completa. Las páginas se podrán pasar pasando el cursor por las esquinas de las imágenes que darán la sensación de levantar, al hacer clic y arrastrar hacia el lado que quiera pasar la página, esta se volteará como si de un libro en físico se tratase. También aparecerá indicado la página en la que se encuentra del total de páginas del libro.

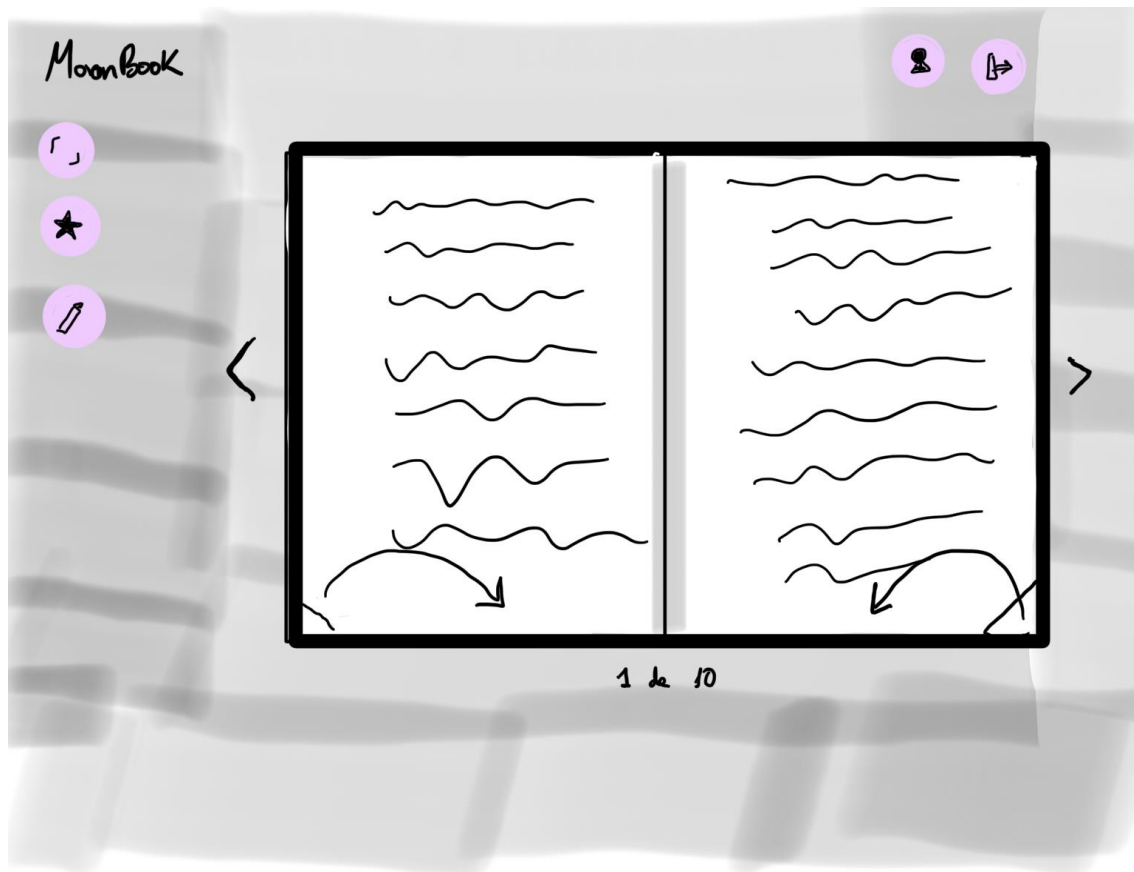


Figura 30 - Mockup visualizar libro

El mockup “Detalles libro” de la Figura 31 satisface el requisito de usuario Tabla 9 - RF-07, desde la página principal de usuario, al pulsar en el icono de información aparecerá a modo de mensaje modal, la información del libro. En la parte derecha superior del contenedor aparecerá un icono con una cruz para cerrar el mensaje modal y en la parte inferior un botón para ir a visualizar el libro.

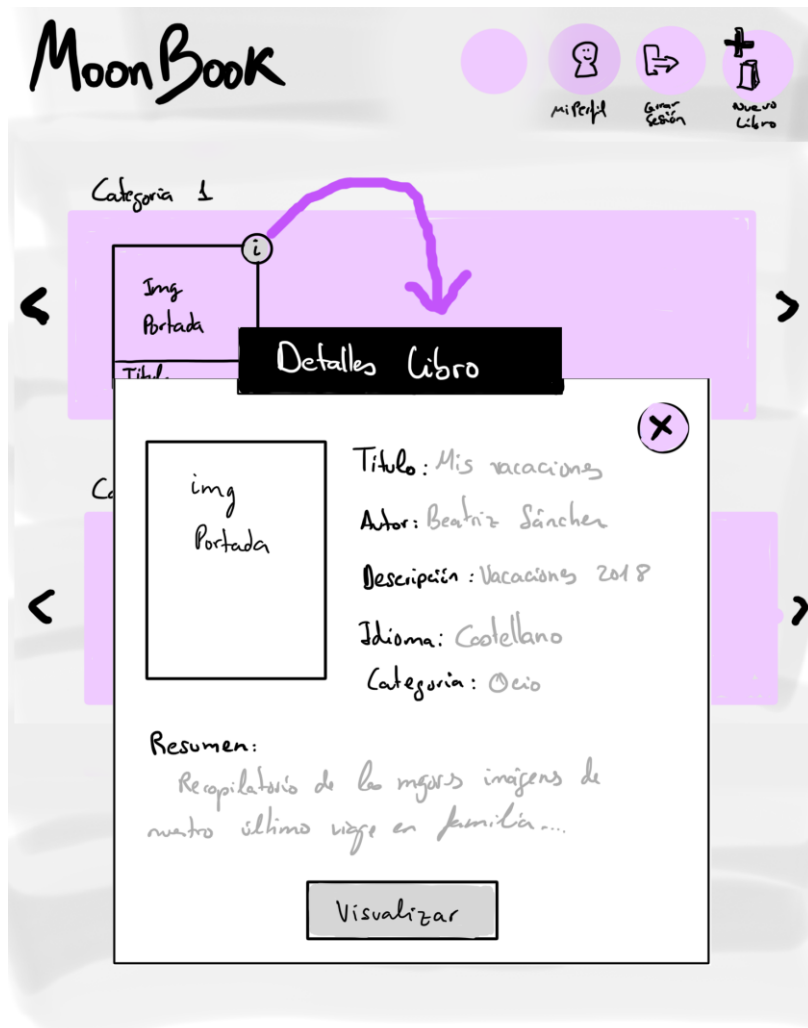
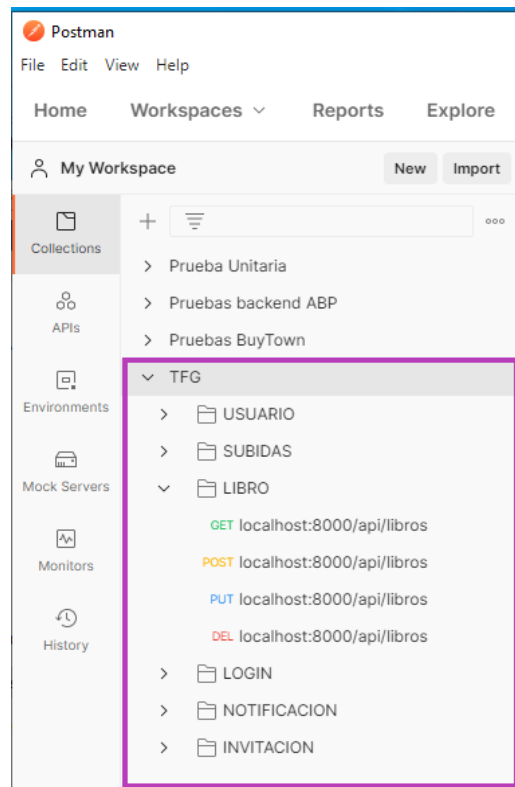


Figura 31 - Mockup información libro

## 8.7. Diseño de pruebas y validación

Para comprobar el funcionamiento correcto de la aplicación se utilizará Postman durante el desarrollo. Postman (24) es una herramienta que nos permitirá comprobar poco a poco el funcionamiento de la API, ya que permite realizar solicitudes REST de manera sencilla y automatizar pruebas. Para cada ruta implementada se realizará una prueba desde esta aplicación.

Para organizar estas peticiones, creamos una colección llamada TFG la cual contendrá varias carpetas como se puede observar en la Figura 32.



*Figura 32 - Postman: estructura de pruebas*

Una vez terminado el desarrollo se comprobarán si se cumplen los requisitos especificados y se mostrará la plataforma a amigos, familiares... para recibir diferentes opiniones que permitan mejorar aspectos de la aplicación.

## 9. Implementación

La implementación de la aplicación se ha realizado por sprints, aportando en cada uno de ellos nuevo valor a la plataforma. En primer lugar, se realizaron pruebas de librerías de visualización del libro y de subida de archivos. Una vez controladas están funcionalidades principales se llevó a cabo el desarrollo completo de la aplicación tanto del backend como del frontend. A continuación, un resumen de los pasos seguidos durante el desarrollo:

### 9.1. Sprint 1 – Comienzo y librería Turn.js

Durante el primer sprint, el primer paso fue elegir el entorno de trabajo, para ello se instaló Visual Studio Code<sup>26</sup>, un editor de código fuente desarrollado por Microsoft, en este caso en un sistema Windows. Este editor facilita la integración con el repositorio de Git, y nos ayuda a escribir bien la sintaxis entre otras características.

El siguiente paso fue descargar ejemplos de la librería Turn.js que proporciona desde la se página web. Una vez funcional el ejemplo, se investigó que hacía cada función y de qué manera se podría hacer que en vez de que el libro estuviera formado por unas páginas estáticas, fuera variable dependiendo del libro que crea el usuario.

Además, se observaron los parámetros que se podían modificar con respecto al tamaño y la visualización de una o dos páginas en la documentación de Turn.js<sup>27</sup>.

### 9.2. Sprint 2 - Archivos PDF

El segundo sprint consistió en transformar un PDF en imágenes. Para ello se utilizó la librería PDF.js<sup>28</sup> de JavaScript que nos permite visualizar un PDF en el navegador, mostrando cada una de esas páginas en una etiqueta canvas de HTML. Pasos seguidos:

1. Obtener fichero adjuntado por el usuario desde un campo de etiqueta input.

---

<sup>26</sup> Visual Studio Code: <https://code.visualstudio.com/download>

<sup>27</sup> Turn.js: <http://www.turnjs.com/#api>

<sup>28</sup> PDF.js: <https://mozilla.github.io/pdf.js/>



2. Abrir fichero con FileReader<sup>29</sup> y pasar a la función `getDocument()` de la librería PDF.js el fichero PDF en formato Uint8Array<sup>30</sup>

3. Una vez cargado, pasando un número a la función `getPage()` de la librería PDF.js nos devuelve la página del PDF .

4. Se define el tamaño del viewport<sup>31</sup> y se llama a la función `render()` pasándole la página deseada del paso 3 y el canvas en el que se va a renderizar.

Una vez que se consiguió visualizar el PDF, el siguiente paso consistió en descargar en formato PNG la página del PDF que se estaba visualizando.

Finalmente, para poder convertir todas las páginas del archivo, internamente, tras descargar una página, se llama a la siguiente y así hasta completar todas las páginas del fichero.

### 9.3. Sprint 3 – Archivos ZIP

En el Sprint 3, ya tenemos un visualizador estático y la posibilidad de adjuntar un PDF y transformarlo en PNG. En el diseño de la aplicación, se especificó que un usuario podría adjuntar un archivo PDF, imágenes, incluso un archivo comprimido ZIP.

Para completar esta funcionalidad, durante este sprint se amplió la posibilidad de subir más de un archivo en el campo input. Se modificó la función que anteriormente recibía un archivo PDF para que, a partir de ahora, por cada uno de los ficheros que se reciben, ya que puede ser más de uno, se compruebe su extensión para realizar un proceso u otro. El esquema de la función queda de la siguiente manera:

-Para cada uno de los archivos:

-Comprobar extensión

-Según extensión:

- PNG/JPEG/JPG: se realizará la subida al servidor
- PDF: se realiza todo el proceso desarrollado en el Sprint 2 y en vez de descargar el PNG, se realizará la subida al servidor

---

<sup>29</sup> FileReader: <https://developer.mozilla.org/es/docs/Web/API/FileReader>

<sup>30</sup> Uint8Array:

[https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\\_Objects/Uint8Array](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Uint8Array)

<sup>31</sup> Viewport: área de pantalla disponible al renderizar un documento.

- ZIP: abrir archivo comprimido con la librería JSZip<sup>32</sup> y realizar el mismo proceso con los archivos que hay en su interior.

## 9.4. Sprint 4 – Backend: comienzo

Una vez desarrolladas las funcionalidades base de la aplicación, comenzó el desarrollo completo de la aplicación tanto del frontend como del backend.

En primer lugar, se instaló MongoDB y se definió la estructura de carpetas:

La carpeta principal se llama TFG y se divide en dos carpetas: backend y frontend. Este Sprint se centró en la carpeta backend. Para comenzar a desarrollar el backend, desde la carpeta con este nombre, creamos un proyecto de Node.js con Express, configuramos la aplicación, que escuchará en el puerto 8000 y se establece la conexión con la base de datos.

También se dejó preparada la organización de carpetas internas:

- Models: directorio que contiene un archivo para cada esquema de Mongo.
- Routers: directorio que contiene un archivo por cada uno de los Modelos existentes, en este fichero se definen las rutas a las que responderá la aplicación.
- Controller: directorio que contiene un archivo por cada uno de los Modelos existentes, en cada fichero se definen las operaciones que se realizarán sobre la base de datos.
- Helpers: directorio que contiene archivos con funciones que puedan ser auxiliares.

Para comprobar el contenido de la base de datos de manera visual, se instaló MongoDB Compass<sup>33</sup>, en la Figura 33 se puede observar algunos de los usuarios almacenados actualmente.

---

<sup>32</sup> JSZip: <https://stuk.github.io/jszip/>

<sup>33</sup> MongoDB Compass: <https://www.mongodb.com/products/compass>

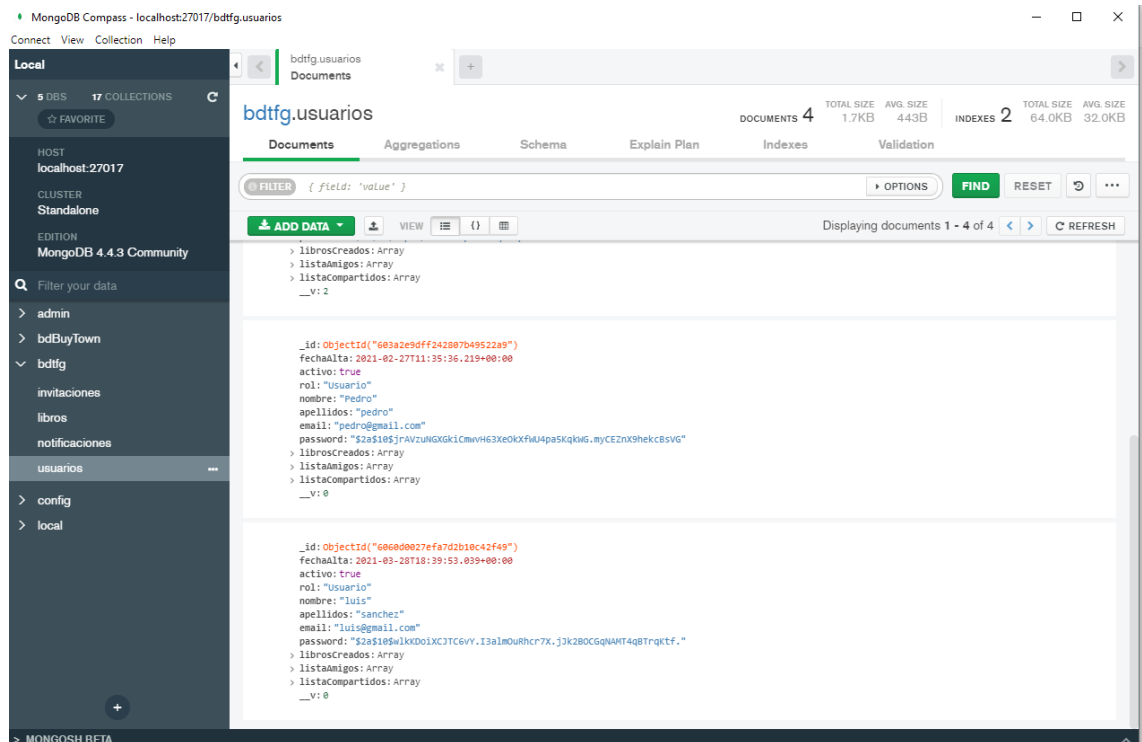


Figura 33 - Captura MongoDB Compass - Usuarios

## 9.5. Sprint 5 – Backend: usuario y libro

En el sprint 5 se especificó el modelo, las rutas y el controlador (crear, editar, eliminar y obtener) tanto de un usuario como de un libro. Destacar que, en la creación de un usuario, en primer lugar, se comprueba si el email que ha introducido el usuario ya existe en la base de datos, de forma que no puede existir en la plataforma dos usuarios con el mismo email y, además, la contraseña se guarda codificada a través de la librería bcrypt<sup>34</sup>. Como se puede observar en la Figura 34, en primer lugar, se genera la sal y a continuación, se crea un hash único pasándole la contraseña del usuario y la sal generada. De esta forma en la base de datos queda guardado el hash generado.

```
const salt = bcrypt.genSaltSync();
const enPassword = bcrypt.hashSync(password, salt);

object.password = enPassword;
```

Figura 34 - Captura código encriptar contraseña usuario

<sup>34</sup> Bcrypt: <https://www.npmjs.com/package/bcrypt>

Una vez programado, con Postman se fue comprobando que funcionaban correctamente las funciones implementadas y que las rutas eran correctas.

Para crear un libro, se establecen los datos que introduce el usuario, quedó pendiente para otro sprint, asignar el libro a cada usuario del listado de amigos si el libro es compartido con amigos.

Para editar un libro, se comprobó que sólo pueda ser editado por su autor y además, para la eliminación, se borran las imágenes que lo forman del servidor antes de eliminar el libro.

## 9.6. Sprint 6 – Backend: login

En el siguiente sprint, se realizó el login de la aplicación, para ello se creó una nueva ruta (/api/login) y su correspondiente controlador.

Esta función login recibe como parámetro el email y la contraseña de un usuario y se realizan las siguientes comprobaciones:

- Comprobar si existe en la base de datos un usuario con el email pasado por parámetro
- Si existe, a través de la función compareSync de la librería bcrypt se compara si la contraseña almacenada en la base de datos recuperada en la petición anterior, coincide con la introducida por el usuario.
- En caso de error en alguno de los pasos anteriores, el login no se puede realizar, devolviendo mensaje de error. En caso afirmativo el usuario estaría logueado correctamente y podría empezar a utilizar la aplicación. La petición devuelve un Token que se almacenará en el lado del cliente.

El token que se devuelve si el login es correcto se genera con la librería jsonwebtoken<sup>35</sup>. A través de la función Sign, pasándole los valores a guardar en el token, en nuestro caso el id del usuario, una clave y la fecha de expiración.

## 9.7. Sprint 7 – Backend: subida y obtención de archivos

Una vez creados el usuario y el libro el siguiente paso fue desarrollar la subida de archivos al servidor para asignarlos a las páginas de un libro. Para ello, se instaló la librería express-

---

<sup>35</sup> Jsonwebtoken: <https://www.npmjs.com/package/jsonwebtoken>

fileupload<sup>36</sup> y se configuran algunos aspectos como el tamaño máximo de los ficheros. A continuación, se creó una nueva ruta y su controlador. En la función de subir un archivo se realizan las siguientes comprobaciones:

- Comprobar si llega algún archivo o de lo contrario llega vacío.
- Comprobar que tiene un tamaño adecuado.
- Comprobar que tiene una extensión válida

Hasta este momento, si alguna de las anteriores comprobaciones no se cumple, se devuelve un mensaje de error, en caso contrario, continuamos con la subida del fichero.

Para establecer el nombre del archivo, se ha utilizado la librería uuidv4<sup>37</sup>, que nos devuelve una cadena aleatoria y que las probabilidades de que se repita es prácticamente nula.

Se requirió del módulo fs de Node que permite acceder al sistema de ficheros para leer, crear, actualizar, eliminar y renombrar archivos.

Una vez tenemos el archivo obtenido de la petición a través de req.files.archivo (dónde archivo es el nombre que le hemos asignado en la petición), con la función archivo.mv, copiamos el archivo en la ruta pasada como parámetro.

Si se produce algún error en la subida del archivo, se informa al usuario, en caso contrario, se modifica el libro al que pertenecerá esa página, añadiendo en su array de páginas, el nombre del nuevo fichero.

Si se produce algún error al actualizar la base de datos, se elimina el archivo del servidor con la función fs.unlinkSync(), en caso contrario, se informa al usuario del éxito de la acción.

Una vez implementada y probada la subida de archivos, se desarrolló la obtención de ellos, para ello, se especifica una nueva ruta y se añade una nueva función. En esta función de obtener archivos que recibe por parámetro el nombre del fichero a buscar y devuelve el archivo, si no lo encuentra, devuelve un mensaje de error al usuario.

---

<sup>36</sup> Express-fileupload: <https://www.npmjs.com/package/express-fileupload>

<sup>37</sup> Uuidv4: <https://www.npmjs.com/package/uuidv4>

## 9.8. Sprint 8 – Frontend: comienzo

Una vez conseguido un backend básico, se comenzó el desarrollo del frontend, para ello se creó desde la carpeta frontend de nuestra aplicación, un proyecto de angular y se instaló bootstrap<sup>38</sup> para ayudar a dar estilo. En angular se utilizan componentes para estructurar la aplicación, archivos de routing para especificar qué componentes se cargan en cada ruta y servicios para comunicar el frontend con el backend.

Se creó el primer componente llamado entrada, que contiene estilos y código HTML de la página de inicio de la aplicación con los dos botones para ir a iniciar sesión y registro.

A continuación, el componente de registro, en el que se creó un formulario con los datos para crear un nuevo usuario y desde el código TypeScript se realizan las comprobaciones iniciales y tras crear un servicio con la primera función crear usuario, se importa al TypeScript y se le pasan los datos introducidos. Esta llamada a la función del servicio se realiza mediante un observable, según el resultado de la petición aparecerá un mensaje de error o se producirá una redirección a iniciar sesión.

Para crear el login, se siguen los mismos pasos que para el registro, se crea un nuevo componente, y una nueva función en el servicio para realizar la petición de login. Esta vez, tras realizar la petición, si ha sido correcto se recibe el token del usuario y su id. Este token se guarda y debe ser comprobado tras cada petición al servidor por seguridad. Se deja pendiente la función de renovar token y comprobar la validez del token para futuros sprints.

Si el login se realiza correctamente, se redirige a la página principal de un usuario logueado.

En la Figura 53 se puede observar el resultado final de los formularios de iniciar sesión y registro.

La página principal de un usuario es un nuevo componente que a su vez carga otros componentes, como el header.

El header es un componente que tiene el logo y el menú de navegación, al estar en un componente separado se puede reutilizar con facilidad y si hay que modificar cuestiones de estilo, solo se producen una vez y no una por cada página en la que aparezca.

Cada archivo TypeScript de un componente, tiene un selector para identificarlo, un templateUrl para cargar el HTML y un styleUrls para especificar el fichero de CSS que cargará. Este selector

---

<sup>38</sup> Bootstrap: <https://getbootstrap.com/>

es el que se utiliza para ser cargado desde otro componente en forma de etiqueta HTML, en este caso `<app-header></app-header>`

Tras el header, se realiza la estructura de la página principal y su diseño mediante HTML y CSS.

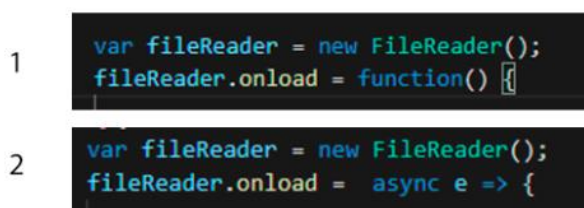
Al final del sprint 8, comienza a tomar forma la aplicación, con una página de entrada básica, queda funcional login y registro y se empieza a formar la página principal de un usuario.

## 9.9. Sprint 9 – Frontend: crear y visualizar libro

Tras iniciar el frontend y tener una base, el sprint 9 consistió en dejar funcional la creación de un libro y su visualización.

La creación del libro se realizó en dos pasos, en primer lugar, un formulario para introducir datos como el título, resumen... y un siguiente paso para adjuntar los archivos.

Para realizarlo se creó un nuevo componente, y se estableció su ruta en el fichero de rutas de angular. El envío del formulario se realizó de la misma forma que los anteriores de login y registro. El paso de adjuntar archivos consistió en adaptar el código realizado en el sprint 2 y 3 de lenguaje JavaScript a TypeScript. La adaptación fue bastante sencilla salvo por el siguiente problema: dentro de la función `onload()` para leer un archivo, no se podía acceder a variables o funciones definidas en el exterior de dicha función. Para resolverlo se utilizó una función de tipo flecha, en la que “this” hace referencia al ámbito externo. En la Figura 35, se puede observar en la primera fila cómo se realizaba en la primera versión y en la segunda fila, el nuevo cambio.



```
1  var fileReader = new FileReader();  
   fileReader.onload = function() {  
2  var fileReader = new FileReader();  
   fileReader.onload = async e => {
```

*Figura 35 - Captura código diferencia función*

Tras este cambio, se quedó funcional la creación de libros a partir de sus datos y los ficheros adjuntados. Se puede ver el resultado final en la Figura 55.

A continuación, se realizó una petición para mostrar en la página principal del usuario los libros que tiene creados. Para realizar la visualización de los libros creados, al pulsar sobre uno de los

libros de la página principal, se realizó la redirección pasando por la ruta el id del libro seleccionado a un nuevo componente que es el encargado de mostrar el libro en formato flipbook.

Desde el componente llamado leer, en un primer momento se instaló la librería de Turn.js al igual que anteriormente con la de PDF.js, sin embargo, resultaron varios problemas y no se consiguió dejar funcional de esta manera. En este punto se estudió cambiar de librería, pero finalmente se optó por guardar en la carpeta assets (carpeta en la que se suelen guardar archivos estáticos, imágenes, videos...) los archivos JS de ejemplo que quedaron funcionales en el sprint 1. Para poder utilizarlos, desde el componente leer, en el archivo de extensión .ts hay que cargarlos y añadirlos a la etiqueta body del HTML. Se obtiene el id del libro a partir de la ruta y se almacena en localStorage<sup>39</sup>. En el sprint 1 se dejó funcional la librería de Turn.js con imágenes estáticas, por lo que en este sprint se realizaron los siguientes cambios para cargar el libro dinámicamente:

- Realizar petición al servidor para obtener los nombres de las imágenes que forman el libro y guardarlos en un array.
- El número de páginas del libro pasará a ser el tamaño del array anterior.
- Se recorre el array y por cada una de sus posiciones se llama a la función cargar página en la que se realiza una petición al servidor para obtener la imagen con el nombre guardado en esa posición.

## 9.10. Sprint 10 – Frontend: Perfil usuario, libros y notificaciones – Backend: notificaciones

Una vez conseguido el funcionamiento básico de la aplicación, poder crear un libro y visualizarlo, se llevó a cabo el perfil del usuario. En primer lugar, se creó un componente menú que se reutilizará dentro del perfil con las diferentes acciones (ver/editar perfil, libros), a continuación, se desarrolló la pestaña ver datos del usuario, que realizando una petición al backend pasándole el id del usuario a buscar nos devuelve sus datos, y para editarlos se hace uso del actualizar usuario creado en el backend en los primeros sprints. Para la pestaña ver/editar/eliminar libros se realizan los mismos pasos que para la de los datos de usuario. En esta pestaña libros solo aparecen los libros creados por el usuario. En la Figura 59 se puede ver

---

<sup>39</sup> LocalStorage: <https://developer.mozilla.org/es/docs/Web/API/Window/localStorage>



el resultado visual de los datos del perfil y el apartado de notificaciones. En la captura izquierda de la Figura 58, se puede observar el listado de libros.

A partir de este sprint, se programa simultáneamente tanto backend como frontend. Se implementa el CRUD básico de una notificación en el servidor, y se añade en el perfil de usuario un nuevo componente para mostrar el listado de notificaciones con la posibilidad de eliminarlas. A su vez se genera una función auxiliar en el servidor para la creación de notificaciones. Esta función será llamada en acciones importantes y generará automáticamente las notificaciones pasándole por parámetro el título y el mensaje. Por ejemplo, se incluye la llamada a crear notificación tras crear un libro correctamente.

### 9.11. Sprint 11 – Backend y Frontend: amigos e invitaciones

Durante el Sprint 11 se implementó en el backend el CRUD de invitaciones y se creó la función añadir amigo en el controlador de usuario.

Desde el frontend, en el menú principal se establece un botón con enlace al nuevo componente añadir amigo. En este componente aparece un formulario con un campo input para introducir el email del amigo. Al rellenar este formulario, se envía el dato introducido a la función creada en el archivo controller del backend. Se puede ver el resultado en la captura izquierda de la Figura 56. Para enviar una invitación de un usuario1 a un usuario2, desde el servidor se realizan las siguientes comprobaciones:

- Comprobar que el email introducido existe en la plataforma.
- Comprobar que el email introducido no está ya en la lista de Amigos del usuario1.
- Comprobar que el email introducido no tiene una invitación enviada anteriormente por el usuario1 en estado pendiente.

Por lo tanto, solo se puede enviar una invitación de amistad, si el usuario2 pertenece a la plataforma, si no pertenece a la lista de amigos del usuario1 o si no tiene invitaciones enviadas en estado pendiente.

En el perfil de usuario, se añaden dos nuevas pestañas:

- Un nuevo componente en el que se muestra el listado de amigos del usuario con la opción de eliminarlo de la lista.
- Otro componente en el que se gestionan las invitaciones tanto las entrantes como las salientes. Las invitaciones pueden estar en tres estados diferentes: pendiente, aceptada o rechazada. Las solicitudes de amistad entrantes en estado pendiente pueden ser

aceptadas o rechazadas. Tras pulsar en una de las dos opciones se actualiza el estado de la invitación desde el servidor y si ha sido aceptada, el usuario entrará en la lista de amigos del usuario que envió la solicitud.

En la Figura 58, la captura central y la de la derecha corresponden al resultado de perfil invitaciones y perfil listado amigos.

En la Figura 36, se puede observar los pasos a seguir de forma resumida para completar la funcionalidad añadir nuevo amigo.

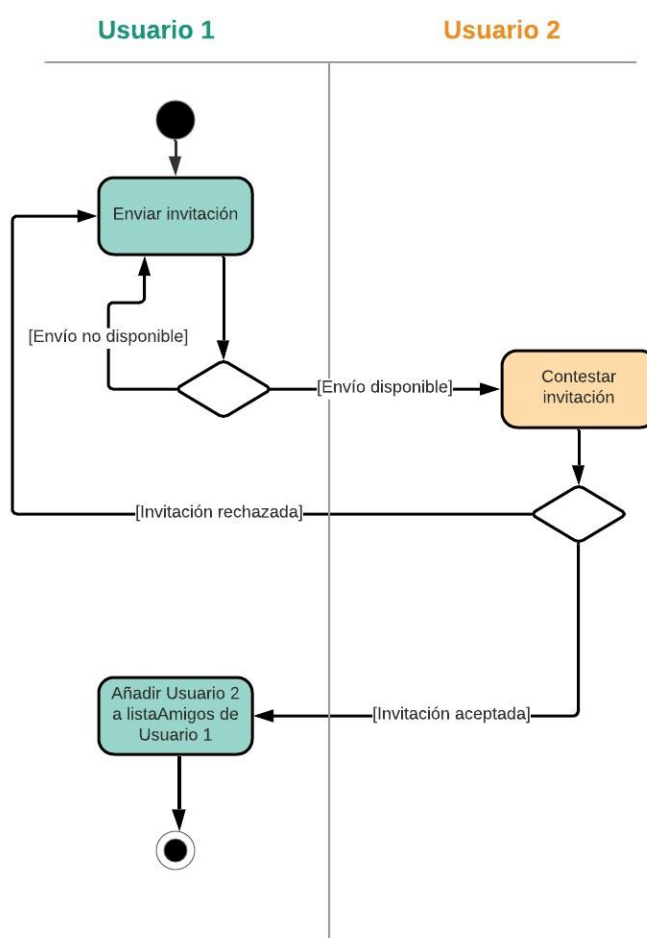


Figura 36 - Diagrama añadir amigo mediante invitación

Tras incorporar la gestión de amigos a la aplicación, se creó una función auxiliar en el backend llamada `compartirLibro` que recibe por parámetro 3 valores (id usuario, id libro y título) y se encarga de asignar al usuario2, el nuevo libro en su array de libros compartidos. Además, se

manda una notificación para informar de la incorporación del nuevo libro. Esta función será llamada en los siguientes casos:

- Al crear un libro. Si el campo compartir tiene el valor amigos, por cada uno de los usuarios del listado de amigos se llamará a la función compartirLibro.
- Al actualizar un libro.

De igual manera, se creó otra función para dejar de compartir un libro con amigos, esta función recibe los mismos parámetros que la anterior y se encargará de buscar el libro a eliminar del array de libros compartidos y de mandar una notificación informando del cambio.

## 9.12. Sprint 12 – Backend y Frontend: ver libros amigos y públicos, detalles libro

Una vez implementada la gestión de invitaciones y modificadas las funciones de crear y editar libros en el servidor, durante el sprint 12 se mostró en la página principal de la aplicación los libros creados por los amigos y los libros compartidos públicamente a la plataforma.

Para ello, se modificó la función del servidor que devuelve los libros creados por un usuario, para que además devuelva el listado de libros que tiene compartidos. Además, en el controlador de libros, se creó una nueva función para devolver todos los libros públicos de la plataforma.

Destacar que, en el apartado de libros públicos, el usuario observará todos los libros públicos de la plataforma excepto los suyos. De esta manera se evita la duplicidad de contenido.

Por cada uno de los libros que aparece en la página principal, se colocó un botón para ver los detalles del libro. Para ello se creó un nuevo componente, y se realizó la petición al servidor para obtener la información de un libro en concreto. El resultado de este nuevo componente se puede ver en la captura derecha de la Figura 56.

Al final de este sprint, se consiguió tener una aplicación con todas las funcionalidades de la aplicación, en la cual ya se pueden crear, ver y compartir libros entre usuarios de la plataforma.

### 9.13. Sprint 13 –Backend y Frontend: últimos libros, opción ver todos y filtrar

En el sprint 13, se realizaron modificaciones en las funciones de obtener libros para limitar el número de resultados a los últimos diez. De esta manera en la página principal, aparecerán los diez últimos libros de cada categoría.

Para ver todos los resultados, se ha incluido en la parte superior derecha de cada fila un botón para mostrar el resto de resultados. En la Figura 54 se puede observar el resultado final de estas interfaces a la izquierda los últimos resultados y a la derecha todos los resultados con la opción del filtrar.

Al pulsar en este botón, se obtienen todos los libros y aparece la opción de filtrar los resultados buscando libros por su título. Para realizar la búsqueda, se ha creado un pequeño formulario con un campo input para que el usuario pueda introducir el texto. Cada vez que se percibe un cambio en el valor de ese input se realiza una nueva petición al servidor para realizar la nueva búsqueda. En la Figura 37 se puede observar como se ha realizado el filtro en la consulta a MongoDB.

```
libro = await Libro.find({$and: [{ autor:libroAutor },{titulo:textoBusqueda}],$or:{jsonCategorias}}
    .populate('autor','-_v -activo -email -fechaAlta -librosCreados -listaAmigos -listaCompartidos -password -rol');
total = await Libro.find({$and: [{ autor:libroAutor }],$or:[{titulo:textoBusqueda}],$or:{jsonCategorias}}).countDocuments();
```

*Figura 37 - Búsqueda en MongoDB*

### 9.14. Sprint 14 –Frontend: zoom y pantalla completa en el visualizador

Una vez que estaban cubiertas las funcionalidades básicas de la aplicación, durante los últimos sprints se han llevado a cabo mejoras, en este sprint se incluyó la posibilidad de hacer zoom en el visor de libros. Para realizar estas funcionalidades, se añadieron dos nuevos botones, que al hacer clic llamarán a unas nuevas funciones añadidas en nuestro código JavaScript.

En esta función, si se ha pulsado en aumentar zoom, la librería Turn.js ofrece la posibilidad de llamar a la función turn() pasando como parámetro la palabra “zoom”, y a continuación el valor a incrementar o disminuir, internamente esta función recalcula el tamaño del libro a partir del valor pasado como parámetro.

Mientras que el zoom esté activado, se podrá hacer scroll horizontal como vertical para moverse entre las páginas y además también se ha implementado la opción de mover el libro haciendo clic y arrastrando, para ello una vez que se detecta clic sobre la etiqueta canvas, mientras se mueve el ratón se calcula su posición actual y anterior para conocer en qué dirección se mueve. En la dirección en la que se producen más cambios, se realizará el scroll.

Visualmente se utilizarán cambios en el puntero, de la siguiente forma:

- Sin zoom: {cursor: default} (cursor en forma de flecha).
- Zoom activado: {cursor: pointer} (cursor en forma de mano).
- Zoom activado y se realiza clic: {cursor: grabbing} (cursor en forma de mano cerrada).
- Zoom activado y se levanta el clic: {cursor: pointer}

Otra funcionalidad añadida es la posibilidad de poner el libro en pantalla completa, para ello se ha añadido un nuevo botón que al pulsarlo llamará a una nueva función creada al igual que las anteriores en nuestro código JavaScript. Desde esta función, obteniendo el elemento a poner en pantalla completa se invoca al método `requestFullscreen()` (según el navegador, esta invocación se realiza de forma distinta). De forma manual cambiamos el tamaño del libro según el tamaño de la pantalla. Al salir del modo pantalla completa, se reducen las dimensiones del libro a su estado anterior. En la Figura 57, se puede observar el resultado final.

## 9.15. Sprint 15 – Frontend: responsive y accesibilidad teclado

Otro aspecto a mejorar en los últimos sprint fue hacer responsive la aplicación web para que se adapte sin problemas a los diferentes tamaños de pantalla. Algunos cambios significativos fueron los siguientes:

En la zona del visor, el libro se adapta a los diferentes tamaños de pantalla capturando el evento `resize`<sup>40</sup>, si se produce cambios en el tamaño de la ventana, el flipbook cambiará la disposición de las páginas de doble a individual o viceversa y además de ajustar las dimensiones de las páginas.

En el perfil de un usuario, cuando la pantalla se hace más pequeña el menú de navegación de la zona izquierda pasa de ser con texto a mostrar iconos.

---

<sup>40</sup> Resize event: [https://developer.mozilla.org/en-US/docs/Web/API/Window/resize\\_event](https://developer.mozilla.org/en-US/docs/Web/API/Window/resize_event)

En toda la aplicación en general, el icono de Moonbook está a la derecha y a la izquierda el menú de navegación principal, cuando se va reduciendo el tamaño de la ventana, ambos elementos se centran quedando en la parte superior el logo y justo debajo el menú.

Se pueden ver estas dos características en la Figura 58 donde en la captura de Perfil libros está en un tamaño de pantalla y la captura de Perfil Invitaciones y amigos está en un tamaño más pequeño de pantalla.

Destacar que se incorporó la accesibilidad por teclado con el atributo tabindex en las diferentes etiquetas HTML.

## 9.16. Sprint 16 – Guía página principal y pruebas de funcionamiento

En la página de entrada antes de iniciar sesión se incorporó un flipbook formado por una guía de uso de la aplicación. De esta forma los usuarios que todavía no han iniciado sesión, pueden utilizar un ejemplo del formato flipbook al mismo tiempo que observan las diferentes funcionalidades que ofrece la aplicación y cómo se utiliza. Se puede ver el resultado final en la Figura 52.

Por último, se fueron realizando diferentes pruebas para validar el resultado de la aplicación al completo.

## 10. Pruebas y validación

Una vez terminada la implementación de MoonBook, se realizaron pruebas con la herramienta Postman para comprobar el funcionamiento del backend y se probaron una a una las diferentes funcionalidades. Finalmente, se realizó una prueba con usuarios para recoger diferentes opiniones que puedan ayudar a mejorar la aplicación en un futuro. Como la aplicación no se encuentra disponible en un servidor web, las pruebas con usuarios se han realizado de manera presencial en un ordenador propio lanzando la aplicación a través de localhost.

Para obtener la opinión de los usuarios, se ha realizado un pequeño cuestionario utilizando Google Forms<sup>41</sup>, una herramienta que permite crear formularios con diferentes tipos de preguntas y compartirla a través de un enlace. Además, los resultados obtenidos se pueden exportar a Excel fácilmente.

La evaluación de la aplicación ha sido realizada por un total de 20 personas de diferentes edades entre 20-29 y 52-55 como se puede ver en la Figura 51. Este tamaño de muestra es adecuado para conocer el estado actual de la aplicación y encontrar los diferentes problemas de usabilidad. Según Jakob Nielsen (25), con 5 usuarios ya se obtienen la mayoría de problemas que puede tener una aplicación y así ha sido, tras ver la reacción de los primeros usuarios utilizando MoonBook se ha observado un patrón en el comportamiento del resto de usuarios.

A la hora de realizar las pruebas, todos los usuarios han tenido en una pantalla izquierda el cuestionario y en otro monitor, la aplicación. Antes de comenzar se les ha explicado en qué consiste la aplicación y cuáles son sus principales características. Tras la explicación, han contestado unas breves preguntas, han utilizado la aplicación y finalmente, han vuelto al cuestionario para realizar un segundo bloque de preguntas. Tras la explicación de en qué consiste la aplicación, se ha observado el comportamiento del usuario sin interactuar con él durante la prueba hasta que ha terminado de enviar el cuestionario y se ha dado por terminada la evaluación.

### 10.1. Resultados cuestionario

A continuación, se muestran las diferentes preguntas con los resultados obtenidos en los diferentes bloques:

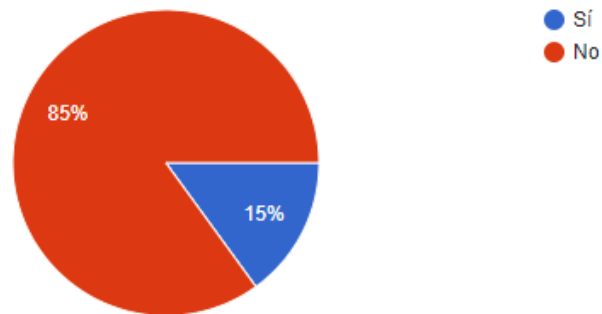
---

<sup>41</sup> Google Forms: <https://forms.gle/1pHzzhcCP28e4F4o6>

- **Bloque 1: antes de utilizar la aplicación:**

¿Conoces el formato "Flipbook" que se menciona en la descripción de esta encuesta?

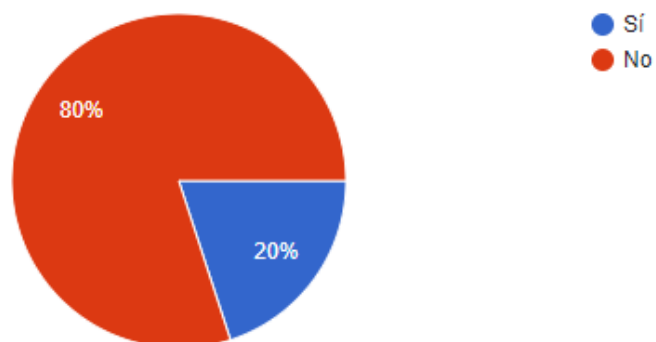
20 respuestas



*Figura 38 - Pregunta: ¿Conoces el formato "Flipbook" que se menciona en la descripción de esta encuesta?*

¿Has utilizado una aplicación parecida a la descrita anteriormente?

20 respuestas



*Figura 39 - Pregunta: ¿Has utilizado una aplicación parecida a la descrita anteriormente?*



En caso de haber utilizado una, ¿De cuál se trata?

4 respuestas

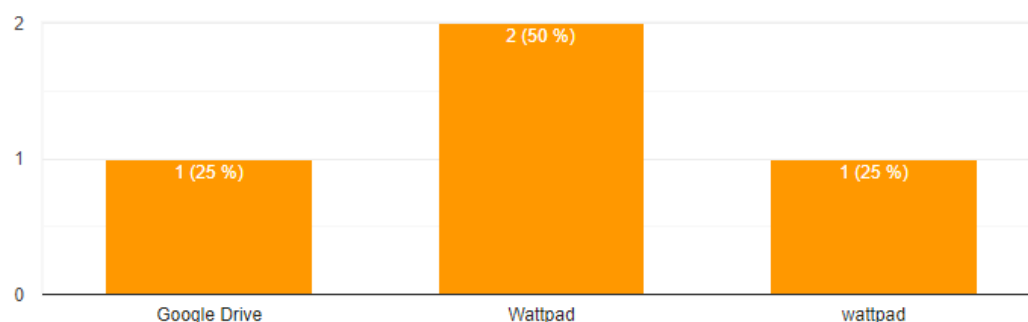


Figura 40 -Pregunta: En caso de haber utilizado una, ¿De cuál se trata?

• **Bloque 2: después de utilizar la aplicación:**

¿Cómo te ha parecido la creación de un libro?

20 respuestas

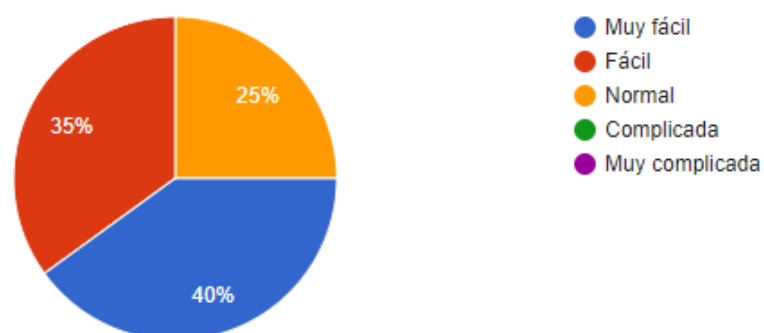


Figura 41 -Pregunta: ¿Cómo te ha parecido la creación de un libro?

¿Cómo te ha parecido el uso del visor de libros?

20 respuestas

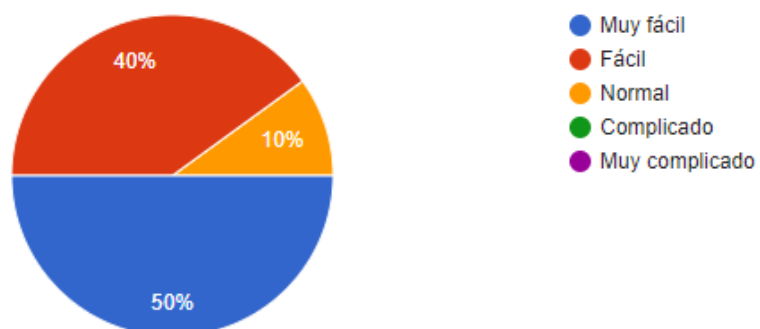


Figura 42 - Pregunta: ¿Cómo te ha parecido el uso del visor de libros?

¿Has conseguido añadir un nuevo amigo?

20 respuestas

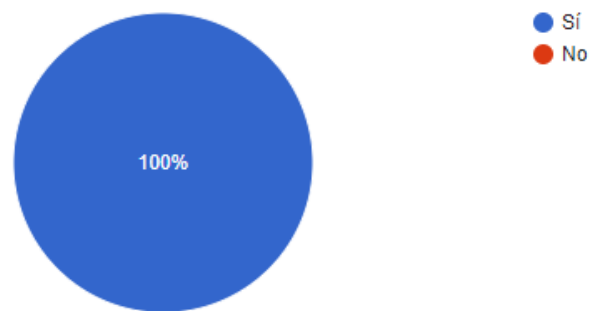


Figura 43 - Pregunta: ¿Has conseguido añadir un nuevo amigo?

¿Has conseguido editar los datos de un libro?

20 respuestas

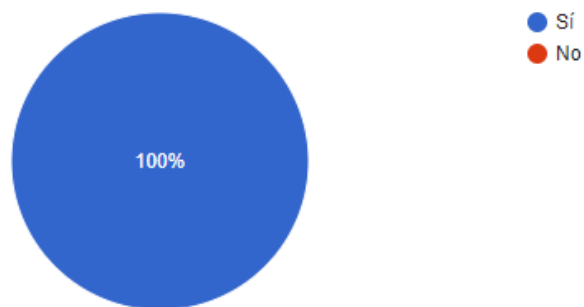


Figura 44 - Pregunta: ¿Has conseguido editar los datos de un libro?

Con respecto al uso del color y tamaño de fuentes y botones ¿Has tenido problemas para visualizar y utilizar de manera correcta la aplicación?

20 respuestas

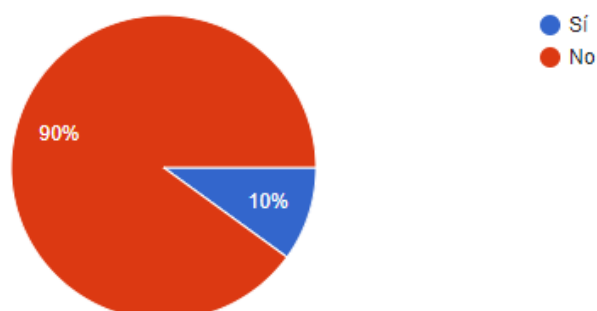


Figura 45 - Pregunta: Con respecto al uso del color y tamaño de fuentes y botones, ¿Has tenido problemas para visualizar y utilizar de manera correcta la aplicación?

De forma general, ¿Qué te ha parecido el uso de la aplicación?

20 respuestas

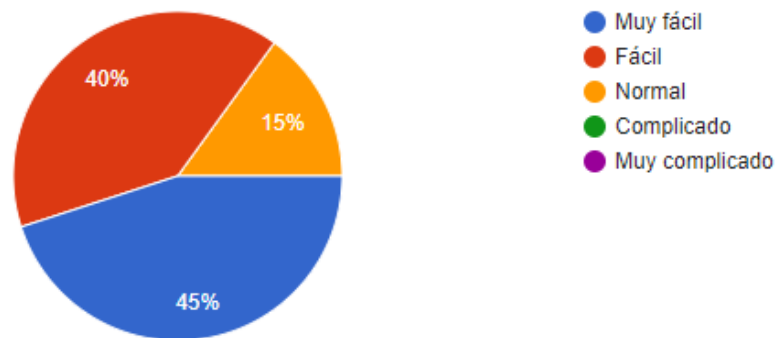


Figura 46 - Pregunta: De forma general, ¿Qué te ha parecido el uso de la aplicación?

¿Qué es lo que más te ha gustado de la aplicación?

20 respuestas

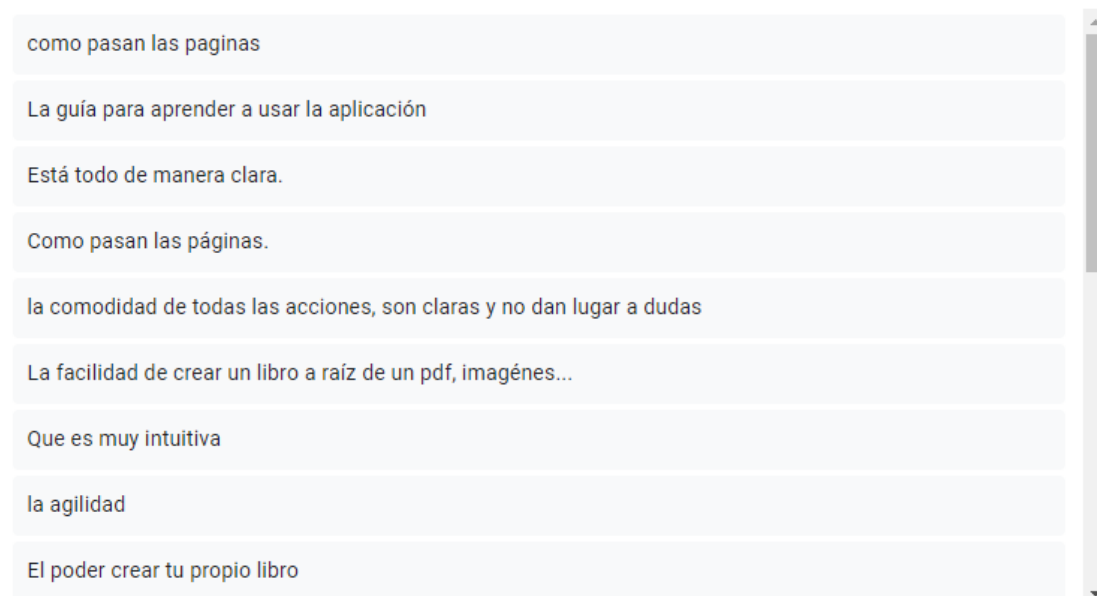


Figura 47 - Pregunta: ¿Qué es lo que más te ha gustado de la aplicación?

¿Qué es lo que menos te ha gustado de la aplicación?

20 respuestas

|   |
|---|
| no se como volver al inicio   |
| no hay texto en el menú   |
| Volver a la pantalla de inicio.   |
| CAMBIAR COLOR PANTALLA DE INICIO  |
| la ubicación del botón de edición de libro  |
| Me ha creado cierta confusión la página principal.  |
| La falta de algunos botones para facilitar el uso en la página, pocos botones pero alguno que sería útil y dificulta un poco el manejo.               |
| Que no sale el mensaje de qué es cada cosa al colocar el cursor encima. No poder editar las imágenes del flipbook en el sitio concreto que se quiere. |

Figura 48 - Pregunta: ¿Qué es lo que menos te ha gustado de la aplicación?

Una vez probada la aplicación, ¿Utilizarías una plataforma para leer que tuviera un visor "flipbook" que simula el uso de un libro en físico ?

20 respuestas

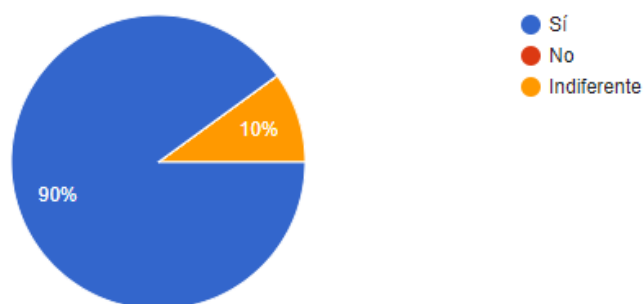


Figura 49 - Pregunta: Una vez probada la aplicación, ¿Utilizarías una plataforma para leer que tuviera un visor "flipbook" que simula el uso de un libro en físico?

En este último apartado puedes dejar cualquier aspecto para ayudar a mejorar la aplicación.

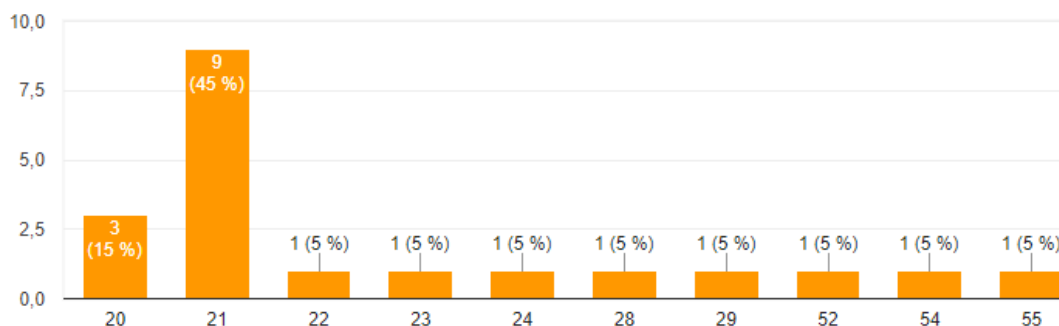
18 respuestas

|  |
|--|
| Me gustaría que tuviera un chat o manera de mandar mensajes a tus amigos   |
| Mejoraría el icono de salida de la aplicación, yo lo colocaría mejor en el apartado del usuario  |
| Si estas leyendo un libro, y por ejemplo no sabes lo que significa, estaría bien poder buscarla directamente desde la aplicación y que salga su significado. Incluso que se pueda tener en la aplicación un listado de palabras aprendidas |
| libro en posición horizontal para fotografías apaisadas.   |
| dar más funcionalidades a los amigos o chat privado en caso de que esté  |
| mejorar la opción de volver al inicio para que quede mas clara con otro icono en la barra de navegación por ejemplo  |
| modificar el contenido del libro   |
| Añadir mensajes de ayuda cuando pones el cursor sobre los iconos. Añadir más categorías y etiquetas de libros. Recomendaciones en función de lo que más consumes.  |

*Figura 50 - Pregunta: En este último apartado, puedes dejar cualquier aspecto para ayudar a mejorar la aplicación*

Edad

20 respuestas



*Figura 51 - Pregunta: Edad*

## 10.2. Conclusiones de la prueba de usuarios

Tras revisar los diferentes resultados podemos obtener las siguientes conclusiones:

- Como se puede ver en la Figura 38, el 85% de los usuarios no conocía en qué consistía el formato flipbook ni había escuchado hablar sobre él.

- Tras conocer de en qué consistía la aplicación, sólo un 20% afirma haber utilizado una aplicación parecida, en este caso Wattpad<sup>42</sup> y otro usuario le ha parecido similar las funcionalidades a Google Drive<sup>43</sup>. Se pueden ver estos resultados en la Figura 39 y en la Figura 40.
- Al preguntar a los usuarios sobre la dificultad a la hora de crear un libro (Figura 41), un 25% lo ha encontrado normal, y el 75% entre fácil o muy fácil.
- Al preguntar sobre la dificultad en el uso del visor de libros (Figura 42), la mitad de los usuarios encuestados lo encuentra muy fácil y un 40% fácil.
- También, se preguntó sobre si fueron capaces de añadir un nuevo amigo (Figura 43) y sobre editar un libro (Figura 44) para ver si eran capaces, de manera autónoma, de encontrar en la aplicación dónde se realizan estas funcionalidades. Todos los usuarios consiguieron completarlas con éxito.
- En cuanto a la Figura 45, se ve como el 90% de los usuarios no tuvieron problemas para visualizar correctamente la aplicación frente a un 10% que comentaron tener problemas para visualizar algunos iconos más pequeños.
- Para extraer una opinión general sobre la dificultad de la aplicación, se realizó la pregunta de la Figura 46, en la que se puede ver que la mayoría de usuarios lo encuentra muy fácil o fácil.
- Como se puede observar en la Figura 47, se realizó una pregunta abierta para conocer qué aspectos son los que más han gustado. Entre ellos destacan los siguientes:
  - El paso de las páginas simulando un libro físico.
  - Facilidad de uso y sencillez en el diseño.
  - La guía de uso inicial.
  - Poder hacer zoom para aumentar el tamaño de las letras en el visor.
- Además, se preguntó sobre los aspectos que menos gustaron de la aplicación (Figura 48). Entre ellos destacan los siguientes:
  - No saber que la página principal era un libro y confundirlo con botones.
  - No saber cómo volver al inicio.
  - Dudas sobre algunos iconos, falta de texto en los iconos al pasar el ratón.
  - Falta de sonido al pasar la página.
  - Dudas sobre cómo se comparten los libros o quién puede acceder a ellos.
  - Falta de chat entre usuarios.

---

<sup>42</sup> Wattpad: [https://www.wattpad.com/?locale=es\\_ES](https://www.wattpad.com/?locale=es_ES)

<sup>43</sup> Google Drive: [https://www.google.com/intl/es\\_es/drive/](https://www.google.com/intl/es_es/drive/)

- Para ver si los usuarios utilizarían una aplicación en la que los libros se visualizan en formato flipbook tras utilizar MoonBook, se realizó la pregunta de la Figura 49, en la que el 90% de los usuarios afirman que la utilizarían.
- Por último, se realizó una pregunta abierta para recibir diferentes mejoras que los usuarios introducirían en MoonBook para su uso diario de la aplicación. A continuación, el listado de mejoras propuestas por los usuarios de manera resumida:
  - Poder ordenar las páginas del libro manualmente.
  - Poder ordenar libros según preferencias del usuario.
  - Introducir modo noche y cambios de color en la interfaz.
  - Introducir chat entre usuarios.
  - Cambiar ubicación icono de salida de la aplicación. Colocarlo como submenú de perfil.
  - Incluir acceso directo a diccionario para poder buscar palabras sin salir del visor y poder añadir estas palabras a un diccionario propio.
  - Cambiar forma de volver al inicio poniendo un botón más en la barra de navegación.
  - Añadir mensajes de ayuda al poner cursor sobre elementos. Poder crear categorías propias. Recomendaciones en función de los libros leídos últimamente.
  - Incorporar bot para poder poner música relajante para acompañar la lectura.
  - Poder seleccionar el archivo que será la portada.
  - Poder compartir el libro a través de un enlace en redes sociales y que los usuarios puedan entrar a través de él con un perfil invitado.
  - Añadir el botón de añadir libro en el visor si es propio para que esté más a la mano.
  - Botones en la interfaz para pasar página de izquierda a derecha.
  - Indicar de una manera más clara que la página principal es interactiva y que se trata ya de un libro

Finalmente, tras analizar los resultados podemos concluir que MoonBook ha resultado fácil de utilizar y lo que más ha llamado la atención de los usuarios es la forma de pasar las páginas. En cuanto a la página principal, la mayoría de los usuarios no se han dado cuenta de que era un libro de guía para utilizar la aplicación, pensando que eran botones. Una vez finalizada experiencia con la aplicación, se les ha enseñado la utilidad de este libro en la pantalla de inicio,

gustándole bastante la idea. Así que, como trabajo futuro, hay que mejorar la presentación de la guía para que quede claro que no son botones y que es un libro interactivo. Otro punto a mejorar es la usabilidad, añadiendo mensajes de ayuda para que el usuario esté seguro de qué acción se realiza al pasar el ratón sobre los diferentes elementos de la aplicación.



## 11. Resultados

A continuación, el resultado final de la aplicación lanzada en localhost:

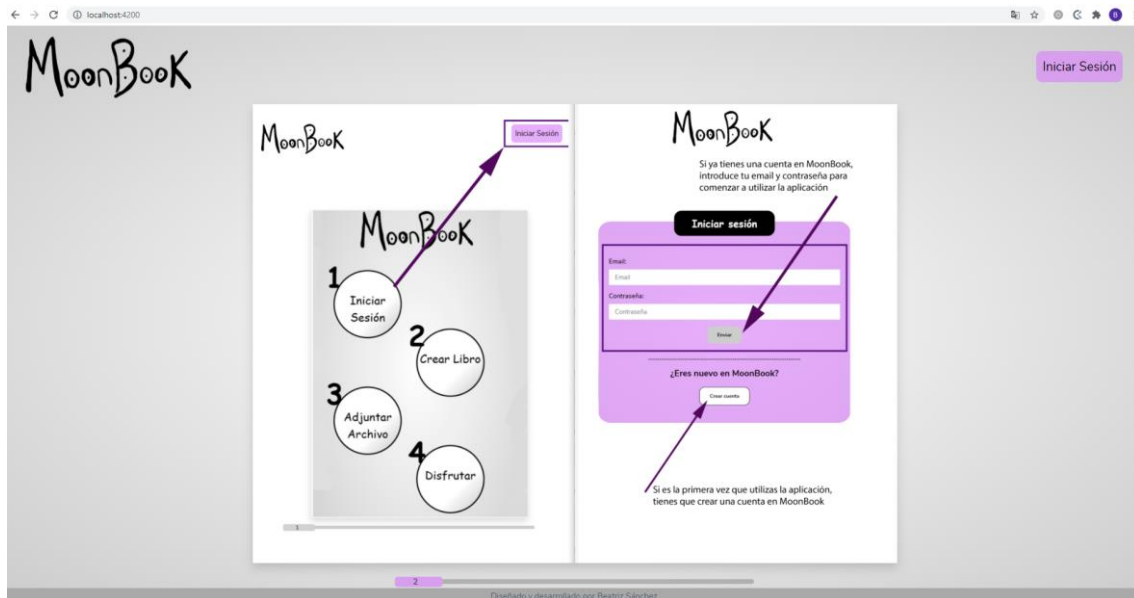


Figura 52 - Resultado final: Página principal

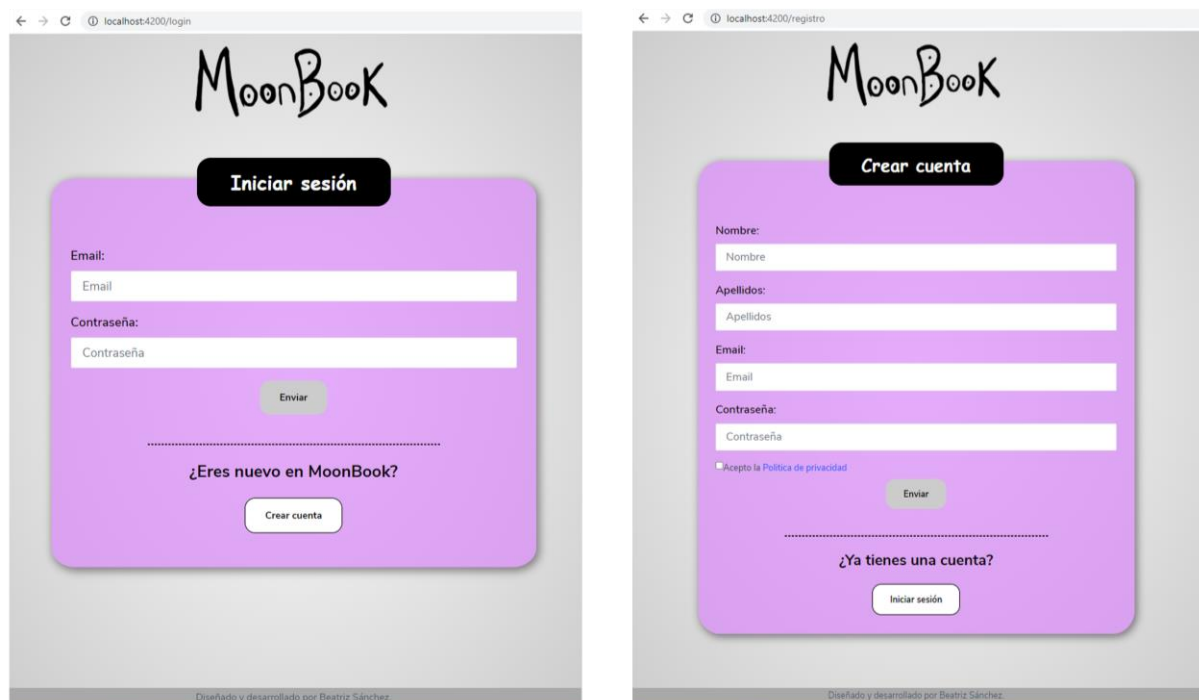


Figura 53 - Resultado final: Iniciar sesión y registro

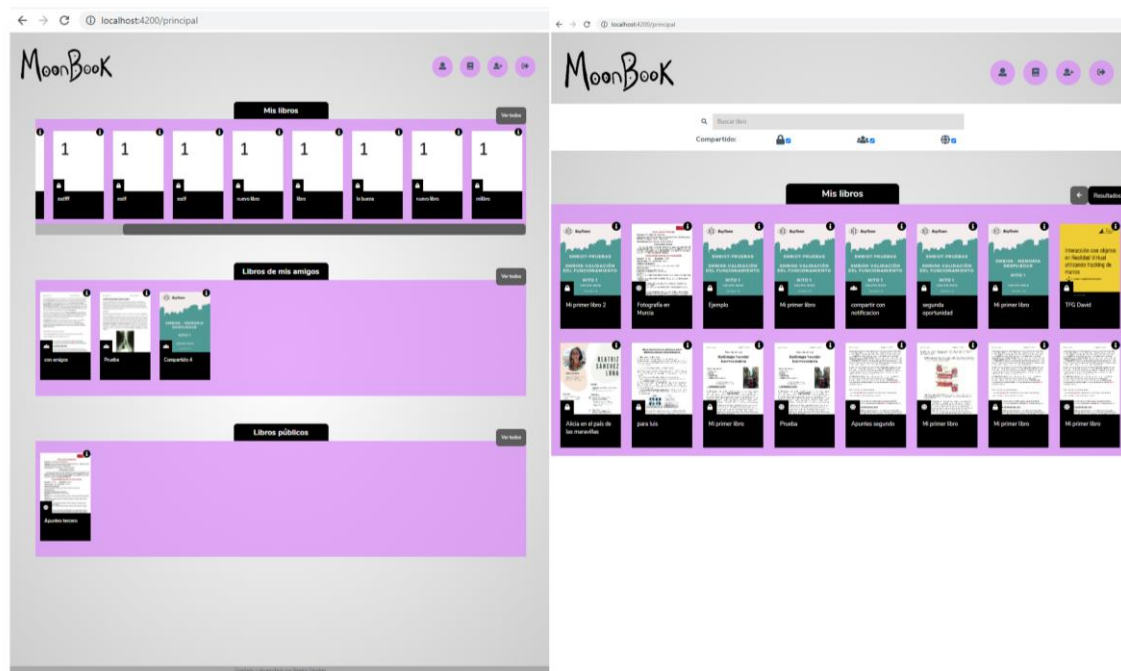


Figura 54 - Resultado final: Página principal y búsqueda

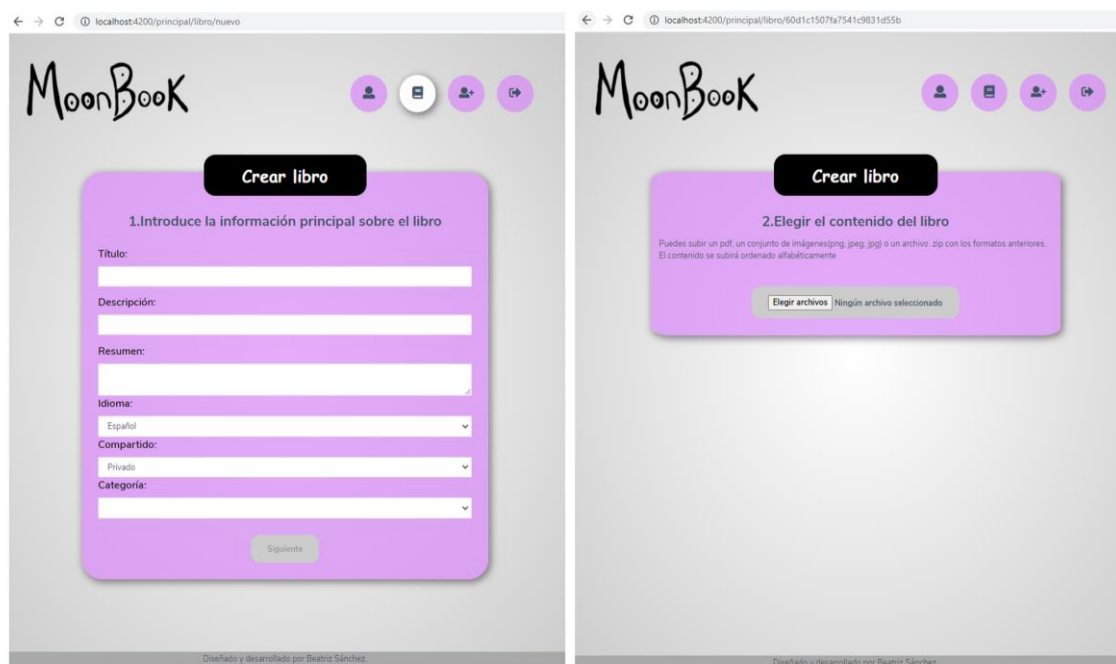


Figura 55 - Resultado final: Crear libro y adjuntar ficheros

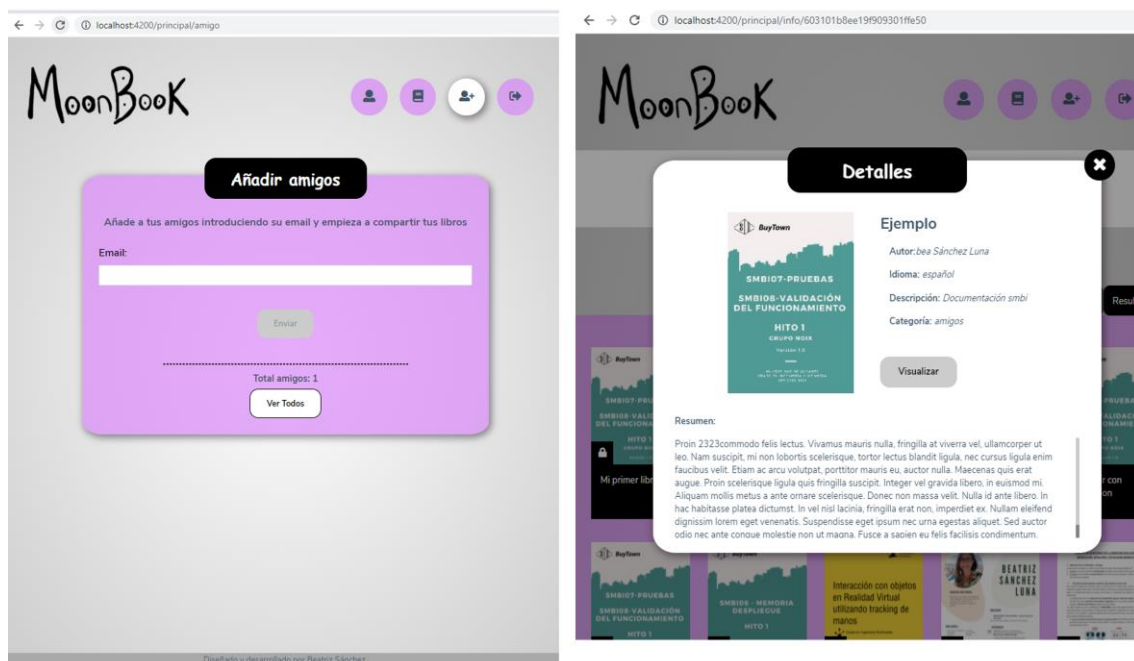


Figura 56 - Resultado final: Añadir amigos y detalles libro

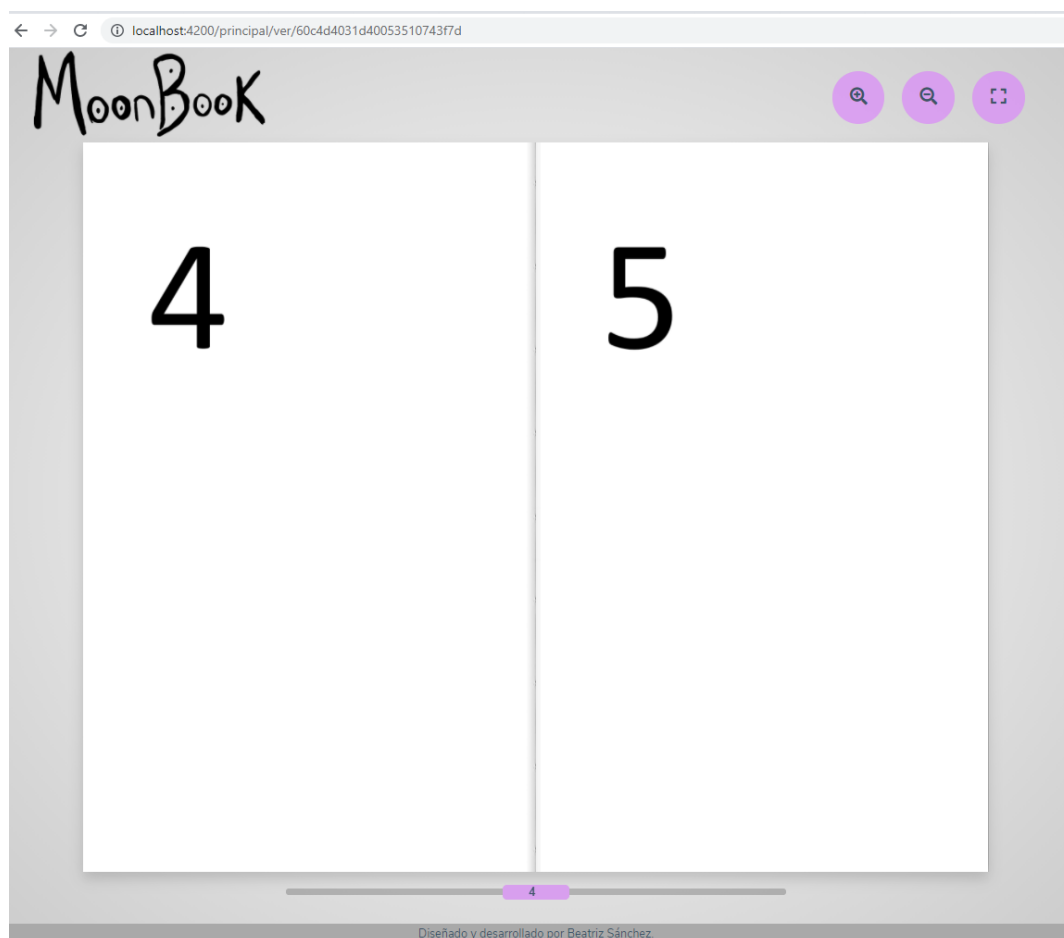


Figura 57 - Resultado final: Visor de libros

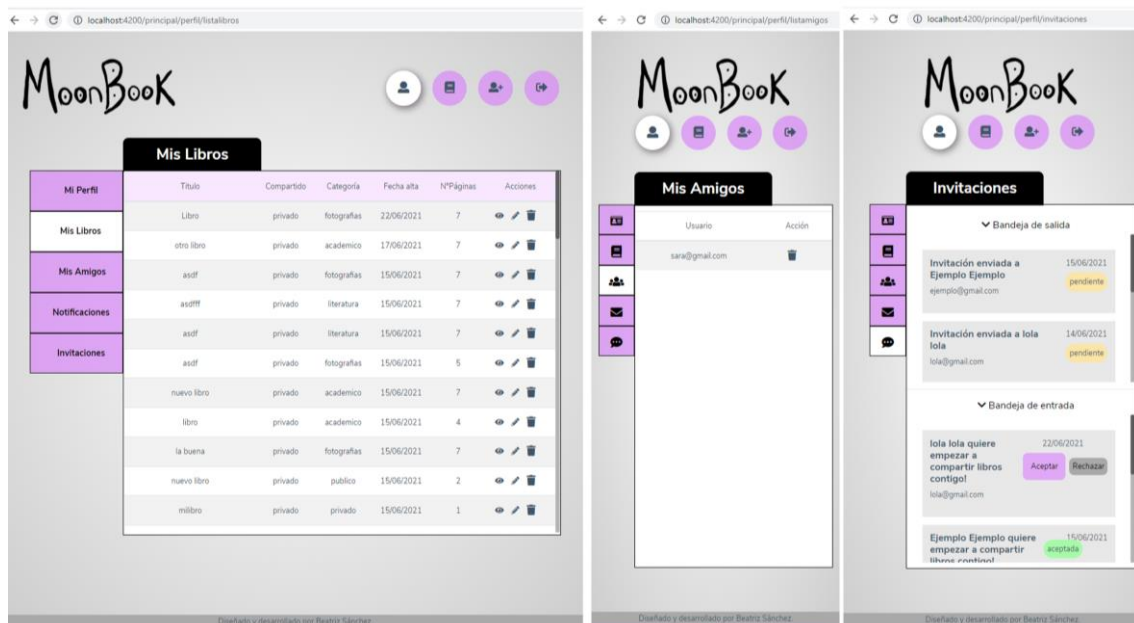


Figura 58 - Resultado final: Perfil libros, perfil amigos y perfil invitaciones

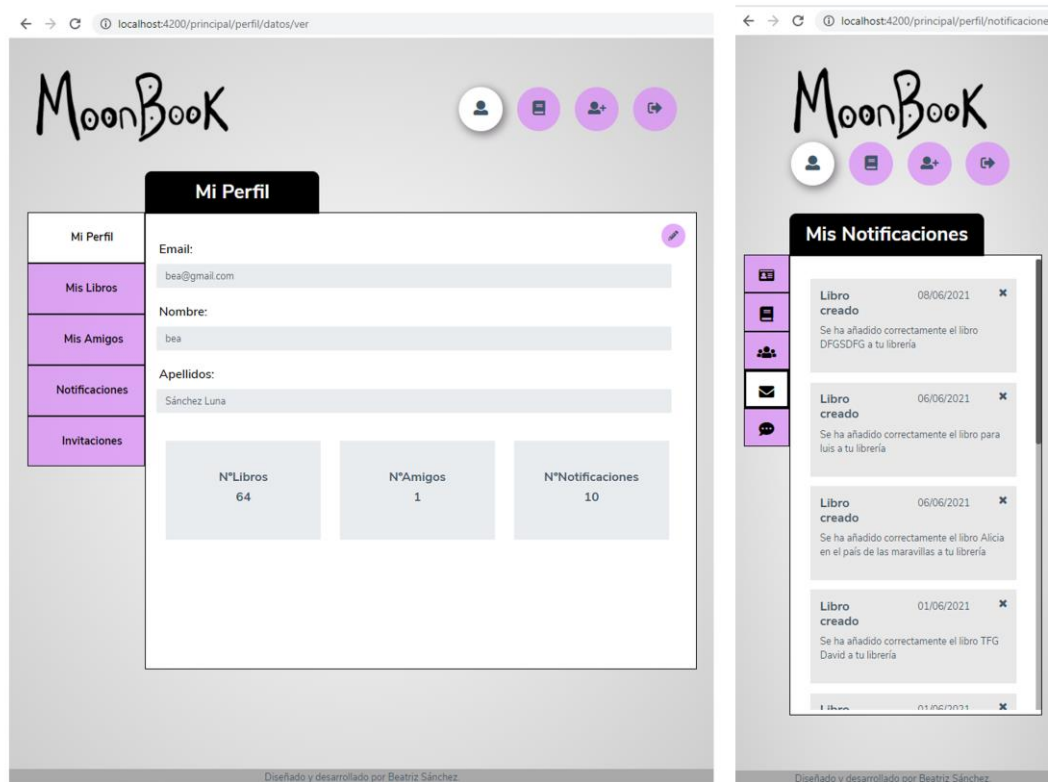


Figura 59 - Resultado final: Perfil usuario y Perfil notificaciones

## 12. Conclusiones y trabajo futuro

Después de todo el TFG, se ha conseguido desarrollar una aplicación que cumple con las funcionalidades diseñadas como adjuntar ficheros y visualizarlos en formato flipbook y además incluir otras funcionalidades que aportan valor a la aplicación como la opción de compartir libros con otros usuarios a través de la plataforma.

Al inicio de este proyecto, no conocía la existencia del formato flipbook para mostrar contenidos a través del navegador y ahora tras estudiar el estado de la cuestión actual he obtenido nuevos conocimientos sobre estas tecnologías y todo lo que se puede llegar a hacer.

Con respecto a los objetivos planteados, se han realizado prácticamente todos los requisitos especificados, teniendo en cuenta tanto la funcionalidad como la maquetación. Solamente ha quedado pendiente la funcionalidad de bloquear a un usuario para que no te pueda mandar más invitaciones de amistad. Puesto que no se contempló en la planificación la subida de la aplicación a la web por tanto el desarrollo y las pruebas de validación con usuarios se han realizado lanzando la aplicación de manera local.

Después de tantos meses de trabajo, aparecen mejoras, nuevas ideas y funcionalidades para añadir en un trabajo futuro que enriquecerían la aplicación:

- Mejorar la seguridad: realizar validez token y seguir comprobando el acceso a las diferentes páginas de la aplicación.
- Mejorar la apariencia y portada del libro guía de la página principal para que los usuarios no tengan dudas de que se trata de un libro en vez de botones.
- Incorporar mensajes de ayuda al pasar el cursor por los diferentes elementos de la aplicación para mejorar la experiencia de usuario.
- Mejorar la accesibilidad para su uso por personas con discapacidad y realizar pruebas con usuarios con discapacidad para evaluar su aceptación.
- Modularizar el conversor de ficheros para que sea más fácil incorporar o eliminar el soporte de diferentes formatos tanto en el frontend como en el backend.
- Alojar la aplicación en un servidor web y realizar la comunicación por Https.
- Permitir la autenticación a través de servicios de terceros con OAuth 2
- Añadir la posibilidad de editar el contenido de un libro (añadir nuevas páginas, eliminar y ordenar).

- Dar al usuario la posibilidad de crear sus propias categorías para ordenar su colección de libros.
- Poder crear un índice a cada libro para que desde el visor se pueda ir directamente a secciones concretas.
- Añadir la opción de realizar comentarios en las páginas de un libro para compartir opiniones, si el autor del libro realiza el comentario pueda elegir su visibilidad (público o privado) de modo que, si es público o compartido con amigos, el resto de usuarios puedan ver y escribir comentarios según la visibilidad, y que el autor del libro pueda eliminar cualquier mensaje asociado a su libro.
- Añadir una papelera de reciclaje para que los libros eliminados permanezcan durante un tiempo ahí para que el usuario tenga la opción de poder recuperarlos
- Añadir paginación en todos los listados para mejorar la carga de grandes volúmenes de información (en listado búsqueda y en los diferentes apartados del perfil de un usuario)
- Añadir funcionalidad comprar libro, sería muy parecida a la funcionalidad existente de añadir amigo con la inclusión de una pasarela de pago.
- Incluir nuevas posibilidades de adjuntar documentos como archivos Word.
- Poder elegir la orientación del libro: vertical o horizontal.
- Poder personalizar el estilo de la aplicación cambiando colores a gusto del usuario.
- Añadir sonidos al pasar página en el visor del libro, incluyendo opción de volumen y mute.
- Añadir nuevos campos por los que filtrar en el buscador, al mismo tiempo que se añaden opciones de búsqueda en los diferentes apartados del perfil.

Por último, después de todo el esfuerzo y trabajo se ha conseguido una aplicación funcional con unas bases estables para seguir creciendo, se han obtenido nuevos conocimientos sobre el formato “flipbook” y se ha adquirido destreza con tecnologías como Angular, NodeJs y MongoDB.

En las últimas semanas de trabajo, se buscó el nombre de MoonBook en internet y aparece una empresa de Madrid ya registrada con el mismo nombre (<https://www.moonbook.es/>), en caso de comercializar este trabajo actualmente académico, saldría al mercado con otro nombre.

## Referencias

1. **recursosenprojectmanagement.** RECURSOS EN PROJECT MANAGEMENT. *Análisis DAFO en proyectos.* [Online] 2021. <https://www.recursosenprojectmanagement.com/>.
2. **Wikipedia, Colaboradores de.** Wikipedia, La enciclopedia libre. *Folioscopio.* [Online] Marzo 24, 2021. <https://es.wikipedia.org/w/index.php?title=Folioscopio&oldid=134240733>.
3. **Trama, Editoras.** Trama Solutions. *¿Qué es un flipbook? Os ponemos un ejemplo.* [Online] Febrero 11, 2016. <https://www.tramasolutions.com/index.php/2016/02/11/que-es-un-flipbook-os-ponemos-un-ejemplo/>.
4. **Flipsnack.** Flipsnack. *Premium.* [Online] 2021. <https://www.flipsnack.com/es/go-premium.html>.
5. **Aps, Paperturn.** Paperturn. *Precios baratos.* [Online] 2020. <https://www.paperturn.com/es/precios>.
6. —. Paperturn. *Prueba gratuita de libro animado.* [Online] <https://www.paperturn.com/es/prueba-gratuita-libro-animado-pdf>.
7. **limited, Wonder idea technology.** FLIPHTML5. *Service and Pricing.* [Online] 2021. <https://fliphtml5.com/fliphtml5-pricing.php>.
8. **Yumpu.com.** Yumpu. *Info-Pricing.* [Online] 2021. [https://www.yumpu.com/xx/info/subscription\\_plans](https://www.yumpu.com/xx/info/subscription_plans).
9. **builder, Flip.** FlipBuilder. *Lista de productos.* [Online] 2020. <http://flipbuilder.es/products.html>.
10. **FlipBuilder.** FlipBuilder. *Flip PDF Professional.* [Online] 2020. <http://flipbuilder.es/flip-pdf-pro/index.html>.
11. **García, Emmanuel.** Turnjs. [Online] 2012. <http://turnjs.com/>.
12. **Riadesign studio, P.Evgenii.** Onebook3d. [Online] 2018. <http://onebook3d.riadesign.ru/>.
13. **UsefulAngle.com.** UsefulAngle. *How to convert PDF to Image (JPEG/PNG) with Javascript using PDF.js.* [Online] Noviembre 16, 2017. <https://usefulangle.com/post/24/pdf-to-jpeg-png-with-pdfjs>.

14. **JSZip.** JSZip. *How to use JSZip.* [Online] <https://stuk.github.io/jszip/documentation/examples.html>.
15. *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software.* **Molina Montero, Bryan , Vite Cevallos, Harry and Dávila Cuesta, Jefferson.** 17, s.l. : Espirales, 2018, Vol. 2.
16. **Robles, Cesar Cid.** Medium. *Comparación bases de datos relacionales y no relacionales.* [Online] Octubre 8, 2019. <https://medium.com/200-response/comparaci%C3%B3n-bases-de-datos-relacionales-y-no-relacionales-424dd83dd693>.
17. **Rendón, Yecid Alexis.** Academia pragma. *Bases de datos relacionales vs. no relacionales.* [Online] Mayo 28, 2019. <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>.
18. **MongoDB.** Mongodb. *La base de datos líder para aplicaciones modernas.* [Online] 2021. <https://www.mongodb.com/es>.
19. **Amodeo, Enrique.** *Principios de diseño de APIs REST.* s.l. : Leanpub, 2013.
20. **StrongLoop.** Expressjs. *Infraestructura de aplicaciones web Node.js.* [Online] Fundación OpenJS. <https://expressjs.com/es/>.
21. **Google.** Angular. *What is Angular?* [Online] 2021. <https://angular.io/guide/what-is-angular>.
22. **Joyent and OpenJS Foundation.** NodeJS. *About.* [Online] <https://nodejs.org/es/about/>.
23. **Montero, Yusef Hassan.** *Experiencia de Usuario: Principios y Métodos.* 2015.
24. **Postman.** Postman. *The Collaboration Platform for API Development.* [Online] 2021. <https://www.postman.com/>.
25. **Nielsen, Jakob.** Por qué solo necesita probar con 5 usuarios. *nngroup.* [Online] Marzo 18, 2000. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.